ArrayAssist Manual

Strand Genomics Pvt. Ltd.
©2006 Strand Genomics. All rights reserved.
Stratagene
©2006 Stratagene. All rights reserved.

Contents

1 ArrayAssist Installation **9**

1.1 Installation on Microsoft Windows 9

1.1.1 Installation and Usage Requirements 9

1.1.2 **ArrayAssist** Installation Procedure for Microsoft Windows 10

1.2 Installation on Linux 12

1.2.1 Installation and Usage Requirements 12

1.2.2 **ArrayAssist** Installation Procedure for Linux 12

1.2.3 Uninstalling **ArrayAssist** from Linux 14

1.3 Installation on Apple Macintosh 14

1.3.1 Installation and Usage Requirements 14

1.3.2 **ArrayAssist** Installation Procedure for Macintosh . . 15

1.4 Installting BRLMM 17

2 ArrayAssist Quick Tour **19**

2.1 **ArrayAssist** User Interface 19

2.1.1 **ArrayAssist** Desktop 19

2.1.2 Desktop Navigator 21

2.1.3 The Workflow Browser 21

2.1.4 The Legend Window 21

2.1.5 Gene List 21

2.1.6 Status Line 24

2.2 Loading Data 24

2.2.1 Loading Data from Files 25

2.2.2 Loading Microarray Data Formats 25

2.3 Projects, Datasets and Views 25

2.3.1 Multiple Projects in **ArrayAssist** 26

2.3.2 Multiple Datasets within a Project 26

2.3.3 Column Type, Attribute and Marks in a Dataset . . . 28

2.3.4 Graphical Views within Datasets . . . 28
2.4 Selecting and Lassoing Rows and Columns . . . 30
2.5 Filtering Data . . . 31
2.6 Algorithms . . . 31
2.7 Data Commands . . . 32
2.7.1 Column Operations . . . 32
2.7.2 Row Operations . . . 34
2.7.3 Dataset Operations . . . 34
2.8 Creating Gene Lists . . . 34
2.9 Tiling Views . . . 37
2.10 Saving Data and Sharing Sessions . . . 37
2.11 The Log Window . . . 38
2.12 Accessing Remote Web Sites . . . 38
2.13 Exporting and Printing Images and Reports . . . 38
2.14 Scripting . . . 39
2.15 Configuration . . . 39
2.16 Getting Help . . . 39

3 Data Visualization **41**
3.1 View . . . 41
3.1.1 View Operations . . . 42
3.2 The Spreadsheet View . . . 49
3.2.1 Spreadsheet Operations . . . 50
3.2.2 Spreadsheet Properties . . . 52
3.3 The Scatter Plot . . . 56
3.3.1 Scatter Plot Operations . . . 57
3.3.2 Scatter Plot Properties . . . 58
3.4 The 3D Scatter Plot . . . 66
3.4.1 3D Scatter Plot Operations . . . 66
3.4.2 3D Scatter Plot Properties . . . 67
3.5 The Profile Plot View . . . 71
3.5.1 Profile Plot Operations . . . 71
3.5.2 Profile Plot Properties . . . 72
3.6 The Heat Map View . . . 76
3.6.1 Heat Map Operations . . . 76
3.6.2 Heat Map Toolbar . . . 81
3.6.3 Heat Map Properties . . . 82
3.7 The Histogram View . . . 86
3.7.1 Histogram Operations . . . 87
3.7.2 Histogram Properties . . . 87

3.8 The Bar Chart . . . 92
3.8.1 Bar Chart Operations . . . 92
3.8.2 Bar Chart Properties . . . 93
3.9 The Matrix Plot View . . . 96
3.9.1 Matrix Plot Operations . . . 98
3.9.2 Matrix Plot Properties . . . 98
3.10 Summary Statistics View . . . 102
3.10.1 Summary Statistics Operations . . . 102
3.10.2 Summary Statistics Properties . . . 104
3.11 The Box Whisker Plot . . . 108
3.11.1 Box Whisker Operations . . . 110
3.11.2 Box Whisker Properties . . . 110
3.12 Trellis . . . 115
3.12.1 Trellis View Operations . . . 116
3.12.2 Trellis Poperties . . . 116
3.13 CatView . . . 118
3.13.1 CatView Operations . . . 118
3.13.2 CatView Poperties . . . 119
3.14 The Lasso View . . . 119
3.14.1 Lasso Properties . . . 119

4 Dataset Operations **125**
4.1 Dataset Operations . . . 125
4.1.1 Column Commands . . . 125
4.1.2 Row Commands . . . 133
4.1.3 Create Subset Dataset . . . 133
4.1.4 Transpose . . . 134

5 Importing Affymetrix Data **137**
5.1 Key Advantages of CEL/CDF files . . . 137
5.2 Creating New Affymetrix Expression Project . . . 138
5.2.1 Selecting CEL/CHP Files . . . 139
5.2.2 Getting Chip Information Packages . . . 139
5.3 Running the Affymetrix Workflow . . . 141
5.3.1 Getting Started . . . 145
5.3.2 Project Setup . . . 145
5.3.3 Primary Analysis . . . 149
5.3.4 CHP/RPT/MAGE-ML Writing . . . 154
5.3.5 Data Transformations . . . 160
5.3.6 Data Exploration . . . 164

5.3.7 Significance Analysis . . . 166
5.3.8 Clustering . . . 172
5.3.9 Save Probeset Lists . . . 173
5.3.10 Import annotations . . . 173
5.3.11 Discovery Steps . . . 173
5.3.12 Genome Browser . . . 176
5.4 Importing CEL/CHP Files from GCOS . . . 177
5.5 Technical Details . . . 178
5.5.1 Probe Summarization Algorithms . . . 178
5.5.2 Computing Absolute Calls . . . 183
5.5.3 GO Computation . . . 183

6 Importing EXON Data **185**
6.1 Analyzing Affymetrix Exon Chips . . . 185
6.1.1 Space Requirements . . . 185
6.2 Importing and Analyzing Exon Data . . . 187
6.2.1 Selecting CEL/CHP Files . . . 187
6.2.2 Getting Chip Information Packages . . . 188
6.3 Running the Affymetrix Exon Workflow . . . 188
6.3.1 Providing Experiment Grouping Information . . . 189
6.3.2 Running Probe Summarization Algorithms . . . 190
6.3.3 DABG Filtering . . . 195
6.3.4 Probeset Statistical Significance Analysis . . . 195
6.3.5 Gene Level Analysis . . . 198
6.3.6 Splicing Index Analysis . . . 200
6.3.7 Views on Splicing Analysis . . . 200
6.3.8 Utilities . . . 202
6.3.9 Summary of Dataset Types in an Exon Project . . . 203
6.3.10 Genome Browser . . . 203
6.4 Algorithm Technical Details . . . 203
6.5 Example Tutorial on Exon Analysis . . . 204

7 Importing Copy Number Data **221**
7.1 Importing Genotyping Data for Copy Number Analysis . . . 221
7.1.1 Selecting CEL Files . . . 221
7.1.2 Getting Chip Information Packages . . . 222
7.2 Running the Copy Number Workflow . . . 223
7.2.1 Providing Experiment Grouping Information . . . 224
7.2.2 Generating Genotype Calls . . . 226
7.2.3 Reference Creation . . . 226

7.2.4 Copy Number and LOH Computation 227
7.2.5 Identify Regions/Genes 229
7.2.6 Import Annotations 230
7.2.7 Genome Browser 230
7.2.8 Space Requirements 230
7.2.9 Algorithm Technical Details 233

8 Analyzing Single-Dye Data **237**
8.1 The Single Dye Import Wizard 238
8.2 The Single-Dye Analysis Workflow 250
8.2.1 Getting Started 251
8.2.2 The Experiment Grouping 251
8.2.3 Primary Analysis 255
8.2.4 Data Viewing 264
8.2.5 Significance Analysis 264
8.2.6 Clustering 271
8.2.7 Save Probeset List 271
8.2.8 Import Gene Annotations 271
8.2.9 Discovery Steps 273
8.2.10 Genome Browser 276

9 Analyzing Two-Dye Data **277**
9.1 The Two Dye Import Wizard 278
9.2 The Two Dye Workflow 289
9.2.1 Getting Started 290
9.2.2 The Experiment Grouping 290
9.2.3 Primary Analysis 293
9.2.4 Data Viewing 310
9.2.5 Significance Analysis 310
9.2.6 Clustering 318
9.2.7 Save Probeset List 318
9.2.8 Import Gene Annotations 320
9.2.9 Discovery Steps 321
9.2.10 Genome Browser 326

10 Annotating Results **327**
10.1 Configuration 328
10.2 Annotation Genes from the Web 330
10.2.1 Marking Annotation Columns 330
10.2.2 Starting Annotation 332

10.2.3 Running an Annotation Workflow . . . 333
10.3 Exploring Results . . . 335
10.3.1 Working with Gene Ontology Terms . . . 335

11 The Genome Browser **341**
11.1 Genome Browser Usage . . . 341

12 Clustering: Identifying Rows with Similar Behavior **349**
12.1 What is Clustering . . . 349
12.2 Clustering Pipeline . . . 350
12.3 Graphical Views of Clustering Analysis Output . . . 351
12.3.1 Cluster Set . . . 351
12.3.2 Dendrogram . . . 357
12.3.3 Similarity Image . . . 364
12.3.4 U Matrix . . . 366
12.4 Distance Measures . . . 368
12.5 K-Means . . . 370
12.6 Hierarchical . . . 371
12.7 Self Organizing Maps (SOM) . . . 373
12.8 Eigen Value Clustering . . . 375
12.9 PCA Clustering . . . 376
12.10 Random Walk . . . 377
12.11 Guidelines for Clustering Operations . . . 378
12.11.1 How to Identify k in K-Means Clustering . . . 378
12.11.2 What is a Recommended Sequence for using Algorithms 379

13 Classification: Learning and Predicting Outcomes **381**
13.1 What is Classification . . . 381
13.2 Classification Pipeline Overview . . . 382
13.2.1 Dataset Orientation . . . 382
13.2.2 Class Labels and Training: . . . 382
13.2.3 Feature Selection: . . . 384
13.2.4 Classification: . . . 385
13.3 Specifying a Class Label Column . . . 385
13.4 Viewing Data for Classification . . . 386
13.4.1 Viewing Data using Scatter Plots and Matrix Plots . . . 386
13.5 Feature Selection . . . 387
13.5.1 ANOVA . . . 387
13.5.2 Kruskal-Wallis Test . . . 388
13.5.3 Saving Features and Creating New Datasets . . . 389

13.5.4 Feature Selection from File . . . 391
13.6 The Three Steps in Classification . . . 391
13.6.1 Validate . . . 392
13.6.2 Train . . . 393
13.6.3 Classify . . . 393
13.7 Decision Trees . . . 393
13.7.1 Decision Tree Train . . . 394
13.7.2 Decision Tree Validate . . . 395
13.8 Neural Network . . . 396
13.8.1 Neural Network Train . . . 397
13.8.2 Neural Network Validate . . . 397
13.9 Support Vector Machines . . . 398
13.9.1 SVM Train . . . 399
13.9.2 SVM Validate . . . 401
13.10 Classification or Predicting Outcomes . . . 401
13.11 Viewing Classification Results . . . 402
13.11.1 Confusion Matrix . . . 402
13.11.2 Classification Model . . . 403
13.11.3 Classification Report . . . 407
13.11.4 Lorenz Curve . . . 409
13.12 Guidelines for Classification Operations . . . 411
13.13 Table of Advantages, Disadvantages of Classification Algorithms . . . 411
13.14 What is the Recommended Sequence of using Algorithms . . . 411
13.15 Typical Cases Explained with Various Views . . . 412

14 Regression: Learning and Predicting Outcomes 417
14.1 What is Regression . . . 417
14.2 Regression Pipeline Overview . . . 417
14.2.1 Dataset Orientation: . . . 417
14.2.2 Class Labels and Training: . . . 418
14.2.3 Feature Selection: . . . 418
14.2.4 Regression: . . . 419
14.3 Specifying a Class Label Column . . . 419
14.4 Selecting features for Regression . . . 420
14.4.1 Correlation . . . 420
14.4.2 Rank Correlation . . . 421
14.5 The Three Steps in Regression . . . 421
14.5.1 Validate . . . 423
14.5.2 Train . . . 424

14.5.3 Prediction 424
14.6 Multivariate Linear Regression 424
14.6.1 Linear Regression Train 424
14.6.2 Linear Regression Validate 430
14.7 Neural Network 430
14.7.1 Neural Network Train 431
14.7.2 Neural Network Validate 434
14.8 Prediction 435
14.8.1 Linear Regression Predict 435
14.8.2 Neural Network Predict 435

15 Principal Component Analysis **437**
15.1 Viewing Data Separation using Principal Component Analysis 437
15.2 Outputs of Principal Components Analysis 438
15.2.1 Principal Eigen Values 438
15.2.2 PCA Scores 438
15.2.3 PCA Loadings 440

16 Statistical Hypothesis Testing and Differential Expression Analysis **443**
16.1 Differential Expression Analysis 443
16.1.1 The Differential Expression Analysis Wizard 444
16.2 Analyzing Non-Replicate Data 454
16.3 Technical Details of Replicate Analysis 455
16.3.1 Statistical Tests 455
16.3.2 Obtaining P-Values 460
16.3.3 Adjusting for Multiple Comparisons 461

17 ArrayAssist Enterprise Client **465**
17.1 Enterprise Server 465
17.2 Setting up the **Enterprise Server** for **ArrayAssist** 467
17.2.1 Setting up Vocabularies for MIAME annotations . . . 468
17.3 Logging in and Logging out of the Enterprise Server 469
17.3.1 Logging into the Enterprise Server 469
17.3.2 Change Password on the **Enterprise Server** 470
17.3.3 Logging out from the Enterprise Server 470
17.4 Accessing the Resources Available on the **Enterprise Server** 471
17.4.1 Browse and Managing the Resources Available on the **Enterprise Server** 471

17.4.2 Open Projects and Access files from the **Enterprise Server** . . . 472
17.4.3 Creating Projects with data files on the **Enterprise Server** . . . 473
17.4.4 Save projects and on the **Enterprise Server** . . . 475
17.4.5 Loading Data Files and Annotations on the Enterprise Server . . . 476
17.5 The Enterprise Explorer . . . 477
17.5.1 Options on Folders on the Explorer . . . 477
17.5.2 Options on Files on the *Enterprise* Explorer . . . 484
17.6 Migrating data from the Gene Traffic Enterprise Server . . . 488
17.6.1 Requirements . . . 491
17.6.2 Preparing for Migration on GT server . . . 491
17.6.3 Preparation for Migration on **ArrayAssist** machine . 492
17.6.4 Running the Migration . . . 493
17.6.5 Post-Migration Cleanups and Restore: . . . 498

18 Scripting **499**
18.1 Introduction . . . 499
18.2 Scripts to Access projects and the Active Datasets **ArrayAssist** . . . 500
18.2.1 List of Project Commands Available in **ArrayAssist** 500
18.2.2 List of Dataset Commands Available in **ArrayAssist** 506
18.2.3 Example Scripts . . . 511
18.3 Scripts for Launching View in **ArrayAssist** . . . 513
18.3.1 List of View Commands Available Through Scripts . . 513
18.3.2 Examples of Launching Views . . . 515
18.4 Scripts for Commands and Algorithms in **ArrayAssist** . . . 518
18.4.1 List of Algorithms and Commands Available Through Scripts . . . 518
18.4.2 Example Scripts to Run Algorithms . . . 524
18.5 Scripts to Create User Interface in **ArrayAssist** . . . 525
18.6 Running R Scripts . . . 528

19 Table of Key Bindings and Mouse Clicks **529**
19.1 Mouse Clicks and their actions . . . 529
19.1.1 Global Mouse Clicks and their actions . . . 529
19.1.2 Some View Specific Mouse Clicks and their Actions . 530
19.2 Key Bindings . . . 530
19.2.1 Global Key Bindings . . . 530

19.2.2 View Specific Key Bindings 531

List of Figures

2.1 **ArrayAssist** Layout 20
2.2 The Workflow Window 22
2.3 The Legend Window 23
2.4 Gene Lists 23
2.5 Status Line 24
2.6 **ArrayAssist** Multiple Project and Associated Tabs 26
2.7 **ArrayAssist** Master and Child Datasets 27
2.8 **ArrayAssist** Views within a Dataset 29
2.9 **ArrayAssist** Append Columns By Formula Dialog 33
2.10 Gene Lists 35
2.11 Gene Lists drop-down menu 36
2.12 Gene Lists drop-down menu 37

3.1 Export submenus 46
3.2 Export Image Dialog 47
3.3 Tools ⟶Options dialog for Export as Image 47
3.4 Error Dialog on Image Export 48
3.5 Menu accessible by Right-Click on the plot views 50
3.6 Spreadsheet 51
3.7 Spreadsheet Properties Dialog 53
3.8 Scatter Plot 57
3.9 Scatter Plot Trellised 59
3.10 Scatter Plot Properties 60
3.11 Viewing Profiles and Error Bars using Scatter Plot 63
3.12 3D Scatter Plot 65
3.13 3D Scatter Plot Properties 68
3.14 Profile Plot 70
3.15 Profile Plot Properties 73
3.16 Heat Map 77
3.17 Export submenus 77

3.18 Export Image Dialog . . . 79
3.19 Error Dialog on Image Export . . . 80
3.20 Heat Map Toolbar . . . 81
3.21 Heat Map Properties . . . 83
3.22 Histogram . . . 86
3.23 Histogram Properties . . . 88
3.24 Bar Chart . . . 91
3.25 Matrix Plot . . . 97
3.26 Matrix Plot Properties . . . 99
3.27 Summary Statistics View . . . 103
3.28 Summary Statistics Properties . . . 105
3.29 Box Whisker Plot . . . 109
3.30 Box Whisker Properties . . . 111
3.31 Trellis of Profile Plot . . . 115
3.32 Trellis Properties . . . 116
3.33 CatView of Scatter Plot . . . 117
3.34 CatView Properties . . . 118
3.35 The Lasso Window . . . 120
3.36 The Lasso Window Properties . . . 121

4.1 Data Menu . . . 126
4.2 Logarithm Command . . . 127
4.3 Absolute Command . . . 128
4.4 Append Column by Grouping . . . 129
4.5 Create New Column by Formula . . . 131
4.6 Import Columns from File . . . 132
4.7 Label Rows . . . 133
4.8 Setting Missing Values . . . 134

5.1 Choose CEL or CHP Files . . . 140
5.2 The Navigator at the Start of the Affymetrix Workflow . . . 141
5.3 The Data Description View . . . 142
5.4 The Affymetrix Workflow Browser . . . 144
5.5 The Experiment Grouping Step in the Affymetrix Workflow Browser . . . 146
5.6 The Experiment Grouping View With Two Factors . . . 147
5.7 Specify Groups within an Experiment Factor . . . 148
5.8 Poly-A Control Profiles . . . 151
5.9 Hybridization Control Profiles . . . 152
5.10 PCA Scores Showing Replicate Groups Separated . . . 153

5.11 Correlation HeatMap Showing Replicate Groups Separated . 154
5.12 CHP Viewer 155
5.13 GCOS Error 156
5.14 Register Sample in GCOS 157
5.15 RPT View 158
5.16 MAGE-ML Error 159
5.17 New Child Dataset Obtained by Log-Transformation 160
5.18 Filter on Calls and Signals Dialog 161
5.19 Variance Stabilization 163
5.20 Reorder Groups for Viewing 165
5.21 Significance Analysis Steps in the Affymetrix Workflow 167
5.22 Navigator Snapshot Showing Significance Analysis Views . . 168
5.23 Statistics Output Dataset for a T-Test 169
5.24 Differential Analysis Report 170
5.25 Filtering 171
5.26 GCOS Error 177

6.1 Specify Groups within an Experiment Factor 191
6.2 Poly-A Control Profiles 193
6.3 Hybridization Control Profiles 194
6.4 Navigator Snapshot Showing Significance Analysis Views . . 196
6.5 Differential Analysis Report 197
6.6 Experimental Grouping for the Colon Cancer Dataset 205
6.7 PCA Scores Plot of the Colon Cancer Dataset 207
6.8 Array Correlations on the Colon Cancer Dataset 208
6.9 Selecting Significant Transcripts 210
6.10 Selecting Significantly Spliced Transcripts 212
6.11 Venn Diagram 213
6.12 The Differential Transcript vs Differential Splicing View . . . 214
6.13 A transcipt showing potential splice variation effects in the Differential Splicing Index along Chromosome View 216
6.14 A transcript showing potential splice variation effects in the Profile Plot Splicing Indices view 217
6.15 Region around potentially alternatively spliced probeset . . . 219

7.1 Specify Groups within an Experiment Factor 225
7.2 Profile Tracks in the Genome Browser 231
7.3 Transition Probabilities for LOH analysis againt Reference HMM 234
7.4 The Paired Normal HMM 235

8.1 Step 1 of Import Wizard . . . 240
8.2 Step 2 of Import Wizard . . . 241
8.3 Step 3 of Import Wizard . . . 242
8.4 Step 4 of Import Wizard . . . 244
8.5 Step 5 of Import Wizard . . . 245
8.6 Step 6 of Import Wizard . . . 249
8.7 The Navigator at the Start of the Single Dye Workflow . . . 250
8.8 The Single Dye Workflow Browser . . . 252
8.9 The Experiment Grouping View With Two Factors . . . 253
8.10 Specify Groups within an Experiment Factor . . . 254
8.11 Normalization . . . 257
8.12 Normalization . . . 257
8.13 PCA Scores Showing Replicate Groups Separated . . . 259
8.14 Correlation HeatMap Showing Replicate Groups Separated . 260
8.15 New Child Dataset Obtained by Log-Transformation . . . 261
8.16 Reorder Groups for Viewing . . . 263
8.17 Significance Analysis Steps in the Singledye Analysis Workflow 265
8.18 Step 1 of Differential Expression Analysis . . . 267
8.19 Step 2 of Differential Expression Analysis . . . 268
8.20 Step 3 of Differential Expression Analysis . . . 269
8.21 Navigator Snapshot Showing Significance Analysis Views . . 270
8.22 Filter on Significance Dialog . . . 271
8.23 GO Browser . . . 274

9.1 Step 1 of Import Wizard . . . 279
9.2 Step 2 of Import Wizard . . . 279
9.3 Step 3 of Import Wizard . . . 281
9.4 Step 4 of Import Wizard . . . 282
9.5 Step 5 of Import Wizard . . . 284
9.6 Step 6 of Import Wizard . . . 288
9.7 The Two-Dye Workflow Browser . . . 291
9.8 The Experiment Grouping View With Two Factors . . . 292
9.9 Specify Groups within an Experiment Factor . . . 293
9.10 Suppress Bad Spots . . . 294
9.11 Background Correction . . . 295
9.12 Normalization . . . 296
9.13 Normalization . . . 297
9.14 MVA Plot . . . 298
9.15 Matrix Plot . . . 299
9.16 PCA Scores Showing Replicate Groups Separated . . . 300

9.17 PCA . . . 301
9.18 New Child Dataset Obtained by Log-Transformation . . . 302
9.19 Filter on Signals . . . 303
9.20 Variance Stabilization . . . 303
9.21 Step 1 of Baseline Transformation . . . 305
9.22 Step 2 of Baseline Transformation . . . 305
9.23 Step 1 of Sample Averages . . . 306
9.24 Step 2 of Sample Averages . . . 307
9.25 Dye Swap Transform . . . 308
9.26 Fill in Missing Values . . . 309
9.27 Combine Replicate Spots . . . 309
9.28 Step 1 of Profile Plot by Groups . . . 311
9.29 Step 2 of Profile Plot by Groups . . . 312
9.30 Step 1 of Differential Expression Analysis . . . 313
9.31 Step 2 of Differential Expression Analysis . . . 314
9.32 Step 3 of Differential Expression Analysis . . . 315
9.33 Differential Expression Report . . . 316
9.34 Volcano Plot . . . 317
9.35 Filter on Significance Dialog . . . 318
9.36 K-means Clustering . . . 319
9.37 Create Probeset List from Selection . . . 320
9.38 Import File . . . 321
9.39 Mark Annotation Columns . . . 322
9.40 Fetch Gene Annotations . . . 323
9.41 GO Browser . . . 325

10.1 Configuring Annotation Database . . . 329
10.2 Mapping Annotation Identifiers . . . 331
10.3 Annotation Dialog . . . 334
10.4 GO Browser Showing Gene Ontology terms for selected genes. 337

11.1 Genome Browser . . . 342
11.2 Tracks Manager . . . 343
11.3 Profile Tracks in the Genome Browser . . . 344
11.4 The KnownGenes Track . . . 345

12.1 Cluster Set from K-Means Clustering Algorithm . . . 351
12.2 Dendrogram of Hierarchical Clustering . . . 356
12.3 Export Image Dialog . . . 359
12.4 Error Dialog on Image Export . . . 360

12.5 Dendrogram Toolbar . . . 361
12.6 Similarity Image from Eigen Value Clustering Algorithm . . . 365
12.7 U Matrix for SOM Clustering Algorithm . . . 367

13.1 Classification Pipeline . . . 383
13.2 Feature Selection Output . . . 389
13.3 Feature Selection Output . . . 390
13.4 Confusion Matrix for Training with Decision Tree . . . 402
13.5 Axis Parallel Decision Tree Model . . . 405
13.6 Neural Network Model . . . 406
13.7 Model Parameters for Support Vector Machines . . . 408
13.8 Decision Tree Classification Report . . . 408
13.9 Lorenz Curve for Neural Network Training . . . 410

14.1 Feature Selection Output . . . 422
14.2 Linear Regression Training Report . . . 425
14.3 Linear Regression Model . . . 426
14.4 Linear Regression Error Model . . . 427
14.5 Neural Network Model . . . 433

15.1 Eigen Value Plot . . . 439
15.2 Scatter Plot of PCA Scores with multi-class data . . . 439
15.3 Scatter Plot of PCA Loadings . . . 440

16.1 Experiment Design . . . 445
16.2 Column Reordering . . . 446
16.3 Analysis Type . . . 447
16.4 Select Test . . . 448
16.5 P-value Computation . . . 451
16.6 Differential Expression Spread-sheet . . . 452
16.7 Differential Expression Analysis Report . . . 453
16.8 Volcano Plot . . . 454

17.1 **ArrayAssist** Layout . . . 466
17.2 Superuser Login Details Dialog . . . 468
17.3 Array Assist Manager Repository setup . . . 469
17.4 The Enterprise Menu on **ArrayAssist** . . . 469
17.5 **Enterprise Server** Login Dialog for Creating aamanager . . . 470
17.6 The *Enterprise* browser in the left panel . . . 472
17.7 Download data files along with the project . . . 473

17.8 Using Data Files for the **Enterprise Server** to Create New Project . . . 474
17.9 Saving project along with data files . . . 475
17.10Enterprise Explorer . . . 478
17.11Right-click menu on a Folder in the *Enterprise* Explorer . . . 478
17.12Right-click menu on a File in the *Enterprise* Explorer . . . 479
17.13The Search menu on Folder Right-Click . . . 479
17.14Advanced Search Dialog . . . 481
17.15Share Dialog on Folders in the *Enterprise* Explorer . . . 482
17.16Property dialog on Folders in Explorer Tree . . . 483
17.17File Versions . . . 485
17.18Annotation View . . . 486
17.19Annotation View . . . 487
17.20Share Dialog on Files in the Explorer . . . 489
17.21Property dialog on Files in Explorer Tree . . . 490
17.22Gene Traffic Migration Intsructions Dialog . . . 494
17.23Gene Traffic Migration Login Dialog . . . 495
17.24Choose Root Repository on **Enterprise Server** . . . 496
17.25Choose Projects for Migration . . . 497
17.26Gene Traffic Migration Report . . . 498

18.1 Scripting Window . . . 500

List of Tables

10.1 **ArrayAssist** Workflows 336
10.2 Web Sites Used for Annotation 340

13.1 Decision Tree Table 394
13.2 Table of Performance of Classification Algorithms 412

16.1 Table of Statistical Tests supported in **ArrayAssist** 449

19.1 Mouse Clicks and their Action 529
19.2 Scatter Plot Mouse Clicks 530
19.3 3D Mouse Clicks 530
19.4 Global Key Bindings 530
19.5 Spreadsheet Key Bindings 531
19.6 Scatter Plot Key Bindings 531
19.7 Histogram Key Bindings 531

Chapter 1

ArrayAssist Installation

This version of **ArrayAssist** is available for Windows, Mac OS X (PowerPC and IntelMac), and Linux. This chapter describes how to install **ArrayAssist** on Windows, Mac OS X and Linux. Note that this version of **ArrayAssist** can coexist with version 3 on the same machine.

1.1 Installation on Microsoft Windows

1.1.1 Installation and Usage Requirements

- Operating System: Microsoft Windows XP, or Windows 2000.
- Pentium 4 with 1.5 GHz and 1 GB RAM for 3' IVT
- Pentium 4 with 2.0 GHz and 2 GB RAM for Exon Array
- Disk space required: 120 MB
- At least 16MB Video Memory. Check this via Start →Settings →Control Panel →Display →Settings tab →Advanced →Adapter tab →Memory Size field. 3D graphics may require more memory. Also changing Display Acceleration settings may be needed to view 3D plots.
- Administrator privileges are required for installation. Once installed, other users can use **ArrayAssist** as well.

1.1.2 ArrayAssist Installation Procedure for Microsoft Windows

ArrayAssist can be installed on any of the Microsoft Windows platforms listed above. To install **ArrayAssist**, follow the instructions given below:

- You must have the installable for your particular platform `arrayAssist40_windows.exe`.
- Run the `arrayassist<edition>_windows.exe` installable file.
- The wizard will guide you through the installation procedure.
- By default, **ArrayAssist** will be installed in the `C:\Program Files\Stratagene\ArrayAssist_4.x_..` directory. You can specify any other installation directory of your choice during the installation process.
- Following this, **ArrayAssist** is installed on your system. By default the **ArrayAssist** icon appears on your desktop and in the programs menu. To start using **ArrayAssist**, you will have to activate your installation by following the steps detailed in the Activation step.

By default, **ArrayAssist** is installed in the programs group with the following utilities:

- **ArrayAssist**, for starting up the **ArrayAssist** tool.
- Documentation, leading to all the documentation available on line in the tool.
- Uninstall, for uninstalling the tool from the system.

Activating your ArrayAssist 4.x

Your **ArrayAssist** installation has to be activated for you to use **ArrayAssist**. **ArrayAssist** imposes a node-locked license, so it can be used only on the machine that it was installed on.

- You should have a valid OrderID to activate **ArrayAssist**. If you do not have an OrderID, register at `http://softwaresolutions.stratagene.com` An OrderID will be e-mailed to you to activate your installation.

- Auto-activate **ArrayAssist** by connecting to **ArrayAssist** website. The first time you start up **ArrayAssist** you will be prompted with the '**ArrayAssist** License Activation' dialog-box. Enter your OrderID in the space provided. This will connect to the **ArrayAssist** website, activate your installation and launch the tool. If you are behind a proxy server, then provide the proxy details in the lower half of this dialog-box. If the autoactivation fails, you will have to manually activate **ArrayAssist** by following the steps given below:

- **Manual activation.** If the auto-activation step has failed, you will have to manually get the activation license file to activate **ArrayAssist**, using the instructions given below:

  - Locate the activation key file `manualActivation.txt` in the `\bin\license\` folder in the installation directory.
  - Go to http://softwaresolutions.stratagene.com/mactivate, enter the OrderID, upload the activation key file, `manualActivation.txt` from the file-path mentioned above, and click Submit. This will generate an activation license file (strand.lic) that will be e-mailed to your registered e-mail address. If you are unable to access the website or have not received the activation license file, send a mail to `techservices@stratagene.com`with the subject `Registration Request`, with `manualActivation.txt` as an attachment. We will generate an activation license file and send it to you within one business day.
  - Once you have got the activation license file, strand.lic, copy the file to your `\bin\license\` subfolder.
  - Restart **ArrayAssist**. This will activate your **ArrayAssist** installation and will launch **ArrayAssist**.
  - If **ArrayAssist** fails to launch and produces an error, please send the error code to `techservices@stratagene.com`with the subject Activation Failure. You should receive a response within one business day.

Uninstalling ArrayAssist from Windows

The Uninstall program is used for uninstalling **ArrayAssist** from the system. Before uninstalling **ArrayAssist**, make sure that the application and any open files from the installation directory are closed.

To start the **ArrayAssist** uninstaller, click Start, choose the Programs option, and select `ArrayAssist4`. Click Uninstall. Alternatively, click Start, select the Settings option, and click Control Panel. Double-click the Add/Remove Programs option. Select `ArrayAssist_4_..` from the list of products. Click Uninstall. The Uninstall **ArrayAssist** wizard displays the features that are to be removed. Click Done to close the Uninstall Complete wizard. **ArrayAssist** will be successfully uninstalled from the Windows system. Some files and folders like log files and data, samples and templates folders that have been created after the installation of **ArrayAssist** would not be removed.

1.2 Installation on Linux

1.2.1 Installation and Usage Requirements

- Linux (i686 libc6 >= 2.2.1)
- Pentium 4 with 1.5 GHz and 1 GB RAM for 3' IVT
- Pentium 4 with 2.0 GHz and 2 GB RAM for Exon Array
- Disk space required: 135 MB
- At least 16MB Video Memory. (Refer section on 3D graphics in FAQ)
- Administrator privileges are NOT required. Only the user who has installed **ArrayAssist** can run it. Multiple installs with different user names are permitted.

1.2.2 ArrayAssist Installation Procedure for Linux

ArrayAssist can be installed on most distributions of Linux. To install **ArrayAssist**, follow the instructions given below:

- You must have the installable for your particular platform `ArrayAssist40_linux.bin`.
- Run the `ArrayAssist40_linux.bin` installable.
- The program will guide you through the installation procedure.
- By default, **ArrayAssist** will be installed in the `$HOME/Stratagene/ArrayAssist_4.x` directory. You can specify any other installation directory of your choice at the specified prompt in the dialog box.

- **ArrayAssist** should be installed as a normal user and only that user will be able to launch the application.

- Following this, **ArrayAssist** is installed in the specified directory on your system. However, it will not be active yet. To start using **ArrayAssist** , you will have to activate your installation by following the steps detailed in the Activation step.

By default, **ArrayAssist** is installed with the following utilities in the **ArrayAssist** directory:

- **ArrayAssist**, for starting up the **ArrayAssist** tool.

- Documentation, leading to all the documentation available online in the tool.

- Uninstall, for uninstalling the tool from the system

Activating your ArrayAssist 4.x

Your **ArrayAssist** installation has to be activated for you to use **ArrayAssist**. **ArrayAssist** imposes a node-locked license, so it can be used only on the machine that it was installed on.

- You should have a valid OrderID to activate **ArrayAssist**. If you do not have an OrderID, register at http://softwaresolutions.stratagene.com An OrderID will be e-mailed to you to activate your installation.

- Auto-activate **ArrayAssist** by connecting to **ArrayAssist** website. The first time you start up **ArrayAssist** you will be prompted with the '**ArrayAssist** License Activation' dialog-box. Enter your OrderID in the space provided. This will connect to the **ArrayAssist** website, activate your installation and launch the tool. If you are behind a proxy server, then provide the proxy details in the lower half of this dialog-box. If the autoactivation fails, you will have to manually activate **ArrayAssist** by following the steps given below:

- **Manual activation.** If the auto-activation step has failed, you will have to manually get the activation license file to activate **ArrayAssist**, using the instructions given below:

- Locate the activation key file `manualActivation.txt` in the `\bin\licence` subfolder of the installation directory.
- Go to http://softwaresolutions.stratagene.com/mactivate, enter the OrderID, upload the activation key file, `manualActivation.txt` from the file-path mentioned above, and click Submit. This will generate an activation license file (strand.lic) that will be e-mailed to your registered e-mail address. If you are unable to access the website or have not received the activation license file, send a mail to `techservices@stratagene.com`with the subject `Registration Request`, with `manualActivation.txt` as an attachment. We will generate an activation license file and send it to you within one business day.
- Once you have got the activation license file, strand.lic, copy the file to your `\bin\license\` subfolder of the installation directory.
- Restart **ArrayAssist**. This will activate your **ArrayAssist** installation and will launch **ArrayAssist**.
- If **ArrayAssist** fails to launch and produces an error, please send the error code to `techservices@stratagene.com`with the subject Activation Failure. You should receive a response within one business day.

1.2.3 Uninstalling ArrayAssist from Linux

Before uninstalling **ArrayAssist**, make sure that the application is closed. To uninstall **ArrayAssist**, run Uninstall from the **ArrayAssist** home directory and follow the instructions on screen.

1.3 Installation on Apple Macintosh

1.3.1 Installation and Usage Requirements

- Mac OS X (10.4 or later)
- Support for PowerPC as well as IntelMac with Universal binaries.
- Processor with 1.5 GHz and 1 GB RAM for 3' IVT
- Processor with 2.0 GHz and 2 GB RAM for Exon Array
- Disk space required: 100 MB

- At least 16MB Video Memory. (Refer section on 3D graphics in FAQ)
- Java version 1.5.0_05 or later; Check using "java -version" on a terminal, if necessary update to the latest JDK by going to Applications →System Prefs →Software Updates (system group).
- **ArrayAssist** should be installed as a normal user and only that user will be able to launch the application.

1.3.2 ArrayAssist Installation Procedure for Macintosh

- You must have the installable for your particular platform `arrayassist<edition>_mac.zip`.
- **ArrayAssist** should be installed as a normal user and only that user will be able to launch the application.
- Uncompress the executable by double clicking on the .zip file. This will create a .app file at the same location. Make sure this file has executable permission.
- Double click on the .app file and start the installation. This will install **ArrayAssist** 4.x on your machine. By default **ArrayAssist** will be installed in
  `$HOME/Applications/Stratagene/ArrayAssist_4.x_` or
  You can install **ArrayAssist** in an alternative location by changing the installation directory.
- To start using **ArrayAssist**, you will have to activate your installation by following the steps detailed in the Activation step.
- Note that **ArrayAssist** is distributed as a node locked license. For this the hostname of the machine should not be changed. If you are using a DHCP server while being connected to be net, you have to set a fixed hostname. To do this, give the command `hostname` at the command prompt during the time of installation. This will return a hostname. And set the `HOSTNAME` in the file /etc/hostconfig to `your_machine_hostname_during_installation`

  For editing this file you should have administrative privileges. Give the following command:
  `sudo vi /etc/hostconfig`
  This will ask for a password. You should give your password and you should change the following line

from
`HOSTNAME=-AUTOMATIC-`
to
`HOSTNAME=your_machine_hostname_during_installation`

- You need to restart the machine for the changes to take effect.

By default, **ArrayAssist** is installed with the following utilities in the **ArrayAssist** directory:

- **ArrayAssist**, for starting up the **ArrayAssist** tool.
- ReportTool, In case the tool refuses to start, run this utility and send the output to `techservices@stratagene.com`for us to troubleshoot the problem.
- Uninstall, for uninstalling the tool from the system

ArrayAssist uses left, right and middle mouse-clicks. On a single button Macintosh mouse, here is how you can emulate these clicks.

- A regular single button click emulates a left click.
- Holding the Apple key down and clicking the mouse emulates a right click.
- Holding the Alt key down and clicking the mouse emulates a middle click.

Activating your ArrayAssist 4.x

Your **ArrayAssist** installation has to be activated for you to use **ArrayAssist**. **ArrayAssist** imposes a node-locked license, so it can be used only on the machine that it was installed on.

- You should have a valid OrderID to activate **ArrayAssist**. If you do not have an OrderID, register at `http://softwaresolutions.stratagene.com` An OrderID will be e-mailed to you to activate your installation.
- Auto-activate **ArrayAssist** by connecting to **ArrayAssist** website. The first time you start up **ArrayAssist** you will be prompted with the '**ArrayAssist** License Activation' dialog-box. Enter your OrderID

in the space provided. This will connect to the **ArrayAssist** website, activate your installation and launch the tool. If you are behind a proxy server, then provide the proxy details in the lower half of this dialog-box. If the autoactivation fails, you will have to manually activate **ArrayAssist** by following the steps given below:

- **Manual activation.** If the auto-activation step has failed, you will have to manually get the activation license file to activate **ArrayAssist**, using the instructions given below:

  - Locate the activation key file `manualActivation.txt` in the `\bin\license\` subfolder of the installation directory.
  - Go to http://softwaresolutions.stratagene.com/mactivate, enter the OrderID, upload the activation key file, `manualActivation.txt` from the file-path mentioned above, and click Submit. This will generate an activation license file (strand.lic) that will be e-mailed to your registered e-mail address. If you are unable to access the website or have not received the activation license file, send a mail to `techservices@stratagene.com`with the subject `Registration Request`, with `manualActivation.txt` as an attachment. We will generate an activation license file and send it to you within one business day.
  - Once you have got the activation license file, strand.lic, copy the file to your `<ARRAYASSIST_INSTALLDIR>\bin\license\`.
  - Restart **ArrayAssist**. This will activate your **ArrayAssist** installation and will launch **ArrayAssist**.
  - If **ArrayAssist** fails to launch and produces an error, please send the error code to `techservices@stratagene.com`with the subject Activation Failure. You should receive a response within one business day.

1.4 Installting BRLMM

In Copy Number Projects, to run the BRLMM algorithm, you will need the Affymetrix BRLMM Analysis Tool available from the affymetrix site. The binaries to run BRLMM on Mac and Linux have been packaged with the tool. However, BRLMM for Windows will have to be independently installed by the user. If BRLMM has not yet been installed on the machine, clicking on the BRLMM link in the Copy Number workflow will

pop-up a dialog requesting the user to install BRLMM. This can be downloaded from http://www.affymetrix.com/support/technical/product_updates/brlmm_algorithm.affx link. The user must register at the http://www.affymetrix.com/ site to download this tool. The downloaded file must be unzipped and the contained EXE file run. The BRLMM Analysis Tool can be installed to any directory and after installation will work directly from **ArrayAssist**.

Chapter 2

ArrayAssist Quick Tour

This chapter gives a brief introduction to **ArrayAssist**, explains the terminology used to refer to various graphical components in the user interface, and provides a high-level overview of the data and analysis paradigms available in **ArrayAssist**.

The description here assumes that **ArrayAssist** has already been installed and activated properly. To install and get **ArrayAssist** running, see Installation.

2.1 ArrayAssist User Interface

A screenshot of **ArrayAssist** with various datasets and views is shown below. The various components of the UI are as follows:

The main window consists of four parts - the Menubar, the Toolbar, the Display Pane and the Status Line. The Display Pane contains several graphical views of the dataset, as well as algorithm results. The Display Pane is divided into three parts:

- The main **ArrayAssist** Desktop in the center, and
- The Navigator, and the Gene List/Legend Window on the left.
- The **ArrayAssist** Workflow Browser, and the Filter dialog on the right.

2.1.1 ArrayAssist Desktop

The desktop accommodates all the views and algorithm results pertaining to each project loaded in **ArrayAssist**. Each window can be manipulated

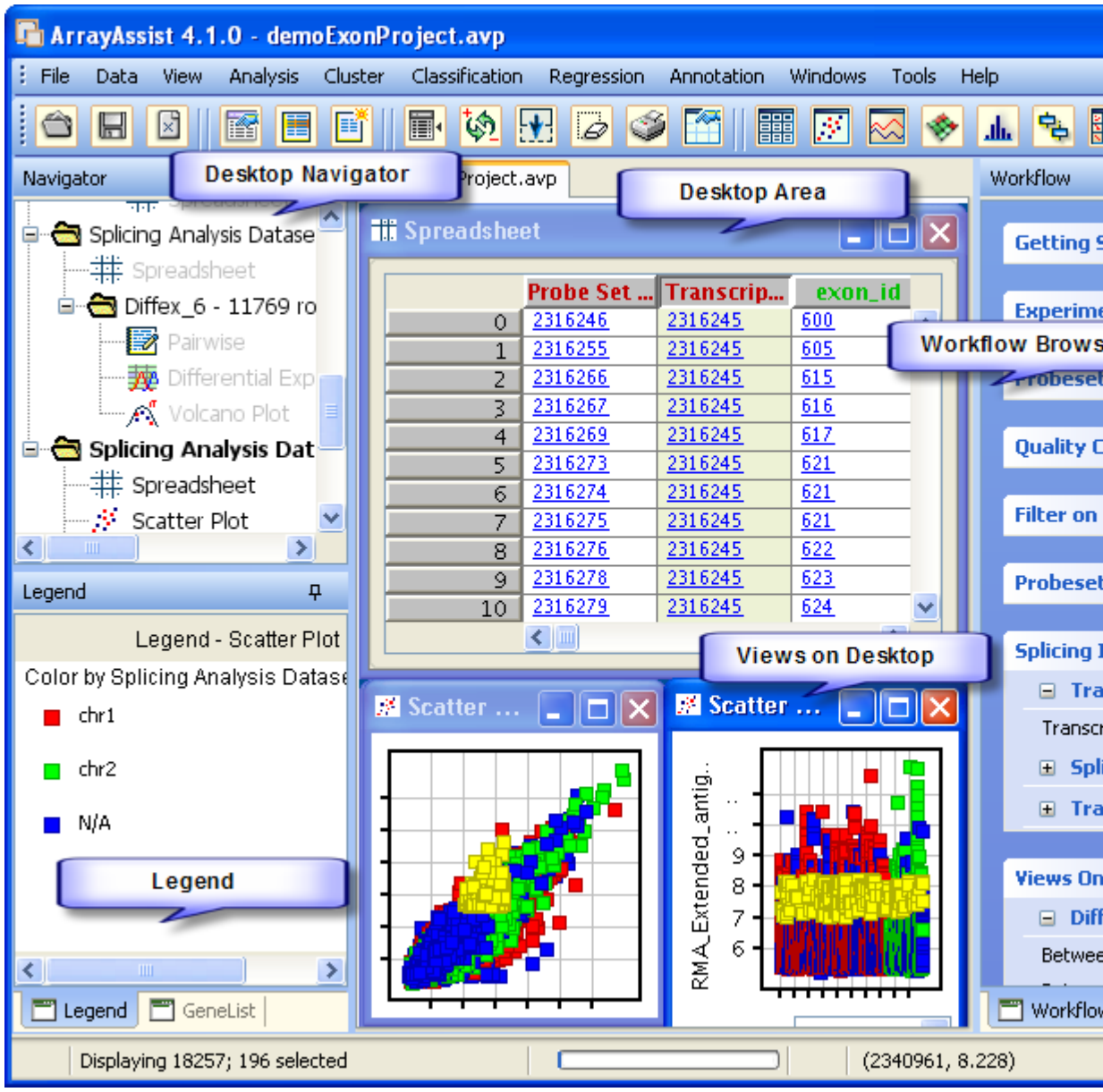


Figure 2.1: **ArrayAssist** Layout

independently to control its size. Less important windows can be minimized or iconised. Windows can be tiled vertically, horizontally or both in the desktop using the *Windows*$\longrightarrow$*Tile* menu.

2.1.2 Desktop Navigator

The desktop navigator displays all currently open datasets, views and algorithm result reports in a hierarchical tree structure. Any of the view windows can be brought into focus by first clicking on the appropriate folder and then clicking on the appropriate icon in the navigator. The navigator window can be resized using the resize bar. It can be completely hidden by clicking on the hide arrow at the top-right of the navigator panel (bottom right on Mac).

Right-clicking on any item in the navigator displays a menu with options to *Delete* the view or to make it *Sticky* (as explained below 2.3.4).

2.1.3 The Workflow Browser

The workflow browser is a key recent addition and allows application specific workflows to appear as a sequence of user clickable links. Each type of project in **ArrayAssist** can potentially have a distinct workflow associated with it.

2.1.4 The Legend Window

The Legend window shows the legend for the current view in focus. Right-Clicking on the legend window shows options to *Copy* or *Export* the legend. Copying the legend will copy it to the Windows clipboard enabling copying into any other Windows application using Control-V. Export will enable saving the legend as an image in one of the standard formats (JPG, PNG, JPEG etc).

2.1.5 Gene List

The Gene List window shows the gene lists that are present in the installation. Gene lists saved from any project is available across all project in **ArrayAssist**. To see the gene lists available in the tool, Right-Click on the GeneList tab in the bottem left of the tool. This will display all the gene lists available in the tool in a tree structure.



Figure 2.2: The Workflow Window

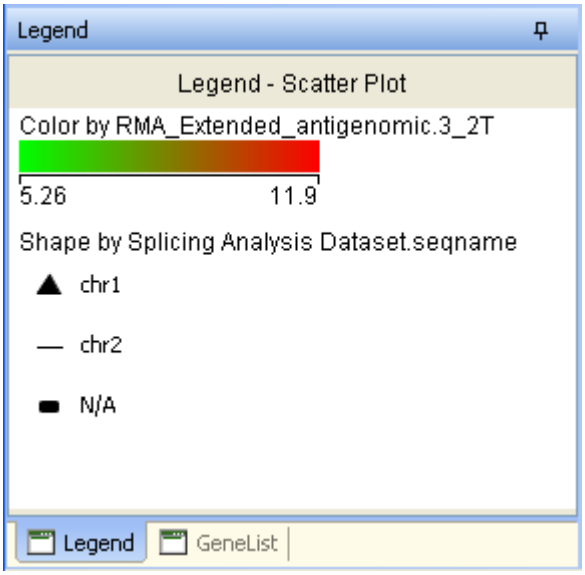


Figure 2.3: The Legend Window

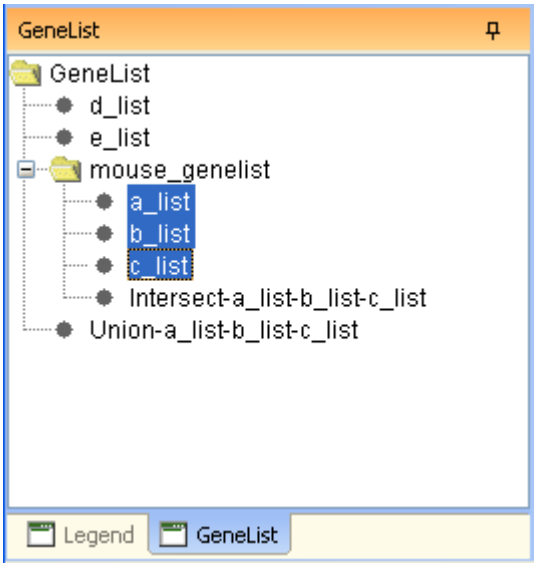


Figure 2.4: Gene Lists



Figure 2.5: Status Line

2.1.6 Status Line

The status line is divided into six logical areas as depicted below.

Status Icon The status of the view is displayed here by an icon. Some views can be in the zoom or the selection mode. The appropriate icon of the current mode of the view is displayed here.

Status Area This area displays high-level information about the current view or algorithm.

Task Progress Bar The progress of the current algorithm/task is displayed in this area as a shaded bar with appropriate information message.

Task Timer displays the time elapsed since the beginning of the current task. Useful to estimate total time required for long running tasks based on the current progress-level and elapsed time.

Ticker Area This area displays transient messages about the current graphical view (e.g., X, Y coordinates in a scatter plot, the axes of the matrix plot, etc.).

Memory Monitor This filed displays the total memory allocated to the Java process and the amount of memory currently used. You can clear memory running the Garbage Collector by clicking on the *garbage Can* icon on the left. This will reduce the memory currently used by the tool.

2.2 Loading Data

Data can be loaded into **ArrayAssist** in multiple ways as briefly outlined below.

2.2.1 Loading Data from Files

Data can be loaded into **ArrayAssist** via the *File →Open* menu or via one of the import wizards. The *File →Open* menu can be used to open tabular text files (comma separated, tab separated or Excel files). In addition, it can also be used to open pre-saved **ArrayAssist** projects with the .avp extension. Somewhat less structured files, like those containing auxiliary lines in addition to tabular data, can also be imported into **ArrayAssist** via the *File →Import Wizard*. This will guide you through importing semi-structured files into **ArrayAssist**. This import wizard also allows users to read data from multiple files and merge them into one dataset.

2.2.2 Loading Microarray Data Formats

ArrayAssist has wizards to read and analyze standard microarray data formats.

New Affymetrix Expression project To start a new project by reading in Affymetrix CEL files, use the *File →New Affymetrix Expression Project* wizard.

New Affymetrix Exon project To start a new project by reading in Affymetrix CEL files, use the *File →New Affymetrix Exon Project* wizard.

New Affymetrix Copy Number project To start a new project by reading in Affymetrix CEL files, use the *File →New Affymetrix Copy Number Project* wizard.

New Single-Dye project To start a new project by loading single-dye files, use the *File →New Single-Dye Project* wizard.

New Two-Dye project To start a new project by by loading two-dye files, use the *File →New Two-Dye Project* wizard.

2.3 Projects, Datasets and Views

Data in **ArrayAssist** is organized into projects. Each project has potentially multiple associated datasets. Each dataset has multiple associated graphical views of the data. This organization into projects, datasets and views is described below in detail.

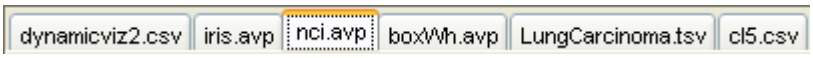


Figure 2.6: **ArrayAssist** Multiple Project and Associated Tabs

2.3.1 Multiple Projects in ArrayAssist

ArrayAssist allows multiple *projects* to be open at the same time. Each project is opened via either the *File →Open* menu (for comma-separated, tab-separated and excel files), the *File→Import Wizard* menu (for one or more files which have a tabular structure embedded inside a non-tabular file, e.g., a file with comment lines), or the *File→New Affymetrix Expression Project* menu (for Affymetrix CEL/CHP files). Each open project has its own display pane and all the available projects are arranged in a multi-tab pane for easy viewing.

2.3.2 Multiple Datasets within a Project

Each project in **ArrayAssist** has a *master dataset* and several other datasets called *child datasets* associated with it. The master dataset contains the original imported data along with all new columns that could have been added in the course of analysis. In addition, it reflects any changes made due to removal or modification of columns.

Child datasets are all derived from the master dataset by taking a subset of rows and columns using *Data→Create Subset→Create Subset from selection*. This hierarchy can go on indefinitely, i.e., one could select rows and columns on a child dataset and then create a further child dataset out of this selection. The latter child dataset will appear nested within the former child dataset on the Navigator as shown in the image below.

Once a child dataset A is created, one could add new columns to this dataset via any of the *Data →Column Commands*. All such columns added to dataset A will appear in A as well as in the master dataset (but not in other datasets between A and the master dataset in the hierarchy). One could also remove columns (*Data→Columns Commands →Remove Columns*) or modify a column in the child dataset (*Data→Row Commands →Label Selected Rows*) or modify the column name or type (via *Data→Data Properties*). In such situations, if this column was derived from a parent dataset, then the change would be effected in the parent dataset as well.

Of all the datasets visible in the Navigator, only one (which appears in bold) will be active at any given time. All others will appear subdued in

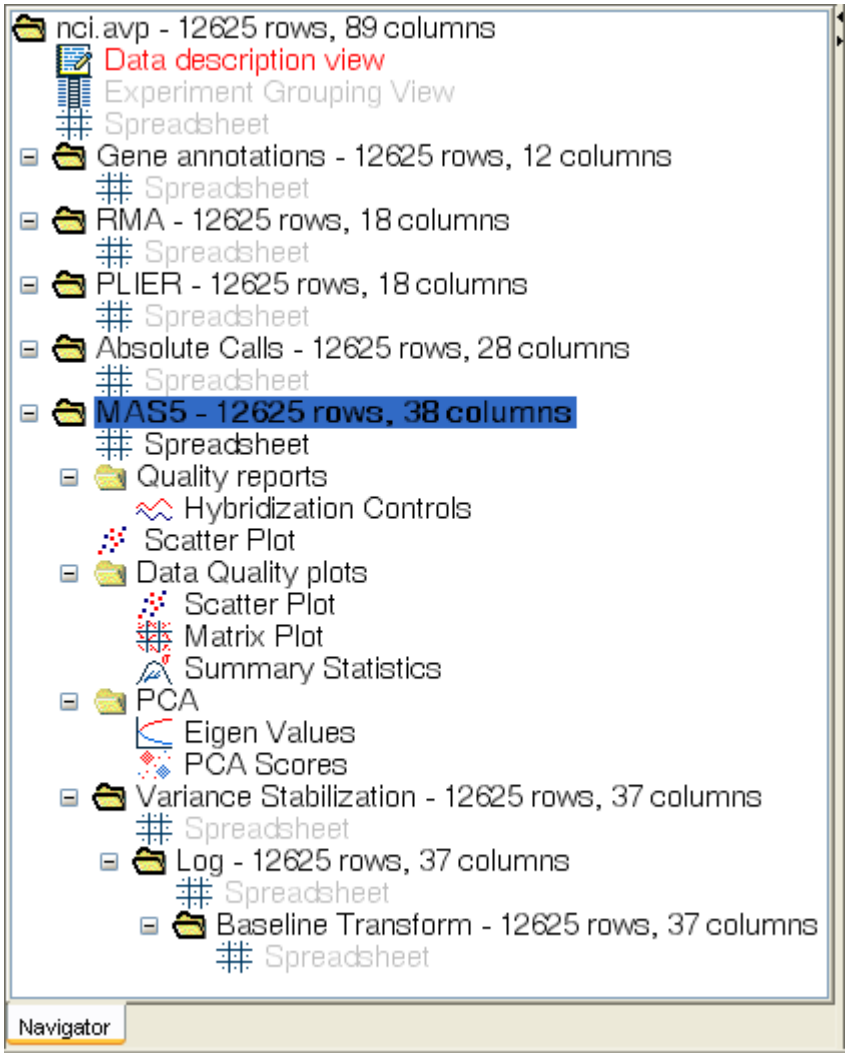


Figure 2.7: **ArrayAssist** Master and Child Datasets

the navigator. To switch datasets, click on the appropriate dataset node in the Navigator.

Row and Column Removal. ArrayAssist does not allow rows to be added or removed from any of the datasets. Only columns can be added and removed.

2.3.3 Column Type, Attribute and Marks in a Dataset

Columns in a dataset have a type (string, float, integer, or date) and a categorical or continuous attribute (decimals are always continuous, strings are always categorical, integers could be either, and dates are always continuous). Column marks denote special column types, e.g., Identifier, URL, Class Label, Locuslink Id etc. Columns marked by one of these marks will be treated in special ways, e.g., marked columns will be automatically copied into child datasets when new child datasets are created, and special features like the Gene Ontology browser will automatically pick up the column marked as Gene Ontology Accession. Column names, types, attributes and marks can be modified using *Data* →*Data Properties*.

2.3.4 Graphical Views within Datasets

From each dataset one can derive various views. These could be direct views available from the *View* menu (like Spreadsheets, Scatter Plots etc) or indirect views obtained by running algorithms like *Clustering* and *Class Prediction* (like Dendrograms). All these views will appear nested within the dataset on the Navigator. Some of these views are table views and are similar in appearance to a dataset spreadsheet. Descriptions of these views appear in Visualization Chapter.

Making Views Sticky. To switch from one view to another within the same dataset, simply click on the view on the Navigator. To switch to a new view within another dataset, move to the other dataset first, and then click on the view. The current active dataset folder will be shown in bold on the navigation tree. To see a view for dataset A within dataset B, go to dataset A and make the view *sticky* by clicking on the view, and using *Right-Click* →*Sticky*. This view will now be available within all other datasets.

Each view is customizable via Right-Click menu options, in particular *Right-Click* →*Properties*.

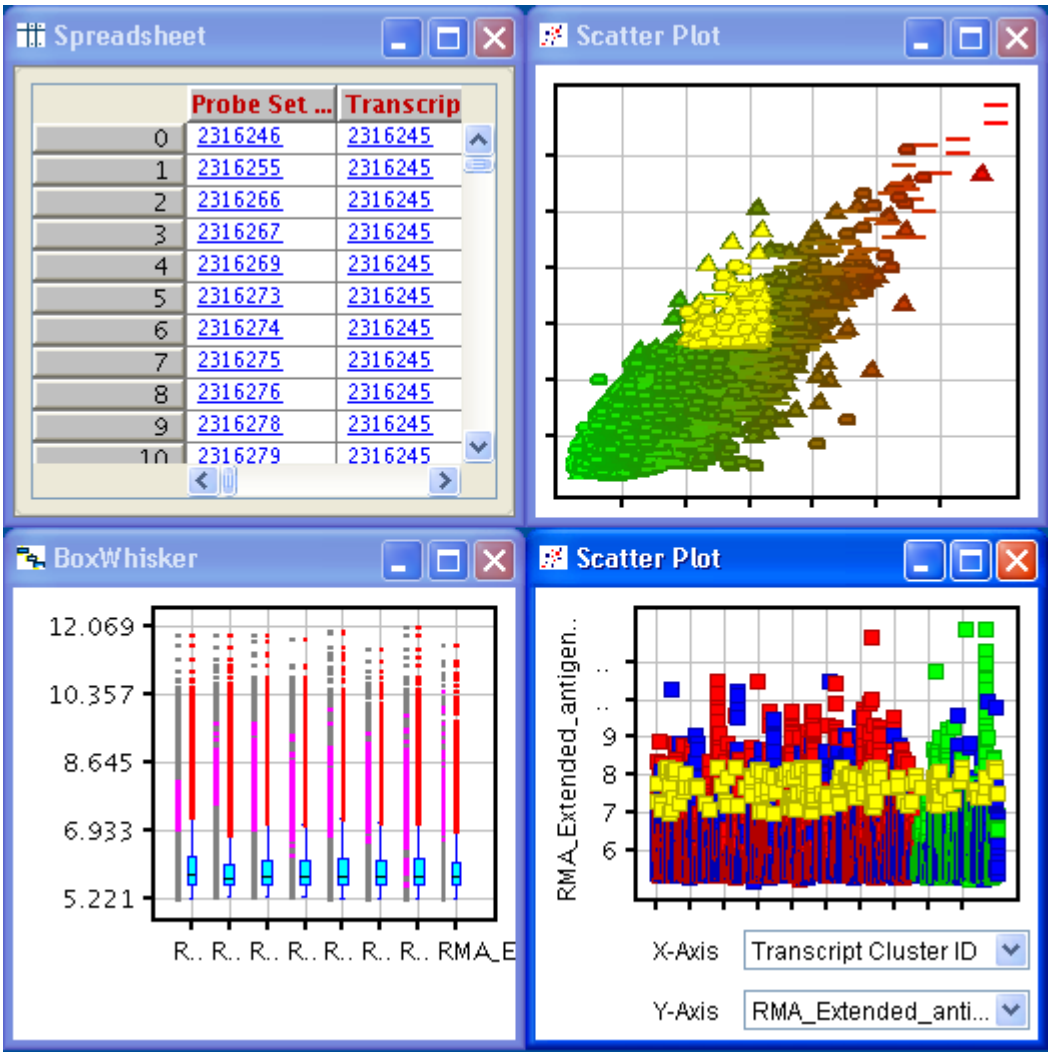


Figure 2.8: **ArrayAssist** Views within a Dataset

2.4 Selecting and Lassoing Rows and Columns

.

Each graphical view allows *subsets of rows* in the data to be selected and highlighted. For example, in a Scatter Plot view, each point corresponds to a row in the dataset. A Left-Click and drag on this view will select all points (i.e., rows) in the region dragged. A distinctive feature of **ArrayAssist** is that these points are highlighted or *lassoed* in all the other open views.

The spreadsheet and other table views in **ArrayAssist** admit both *row selection and column selection*. Rows are selected by clicking on the row headers in the spreadsheet while columns are selected by clicking on column body (and not the header). Clicking on the column header sorts the column (first click sorts in ascending order, second click sorts in the descending order, and the third click restores the original order). Selected rows are lassoed in all the open views while selected columns are highlighted in all open spreadsheets as well as some column based views like the heatmap.

One of the purposes of column selection is to provide selective input to the various views and algorithms and data transformation options available in **ArrayAssist**. Note that all of these algorithms and all the data transformations in *Data→Column Commands* run on all the rows of the spreadsheet but only on the selected columns. This column selection can be performed either in the spreadsheet, or more directly, in the *Columns* tab of the dialog window corresponding to each algorithm/transformation. If no columns are selected, then by default all appropriate columns will be shown as selected in the Columns tab of the dialog window.

Selecting with a Mouse. ArrayAssist uniformly uses the following convention at several places for selection. Left-Click selects the first item (i.e., row, point, etc depending upon the view), Ctrl-Left-Click selects subsequent items and Shift-Left-Click selects a consecutive set of items (in views where contiguity is well-defined). Control-A typically plays the role of Select-All (e.g., on the spreadsheet it selects all columns)

The *Lasso* window available from *View→Lasso* or from the Lasso  icon shows actual data details of the rows selected in any view. Columns in this window can be stretched or shuffled and this configuration is maintained as various selections are performed, allowing the user to concentrate on values in the columns of interest.

Further, **ArrayAssist** supports a special column mark called the URL that can be set from *Data →Data Properties*. Double-Clicking on a URL cell in the spreadsheet or the Lasso window will open that URL in a browser. Note that **ArrayAssist** does not have a column lasso window, i.e., only selected rows are showed in the lasso, not the selected columns. In addition, the Lasso view itself does not allow any selection.

2.5 Filtering Data

ArrayAssist allows filtering of data by setting subranges for columns values in any of the datasets. This is done by using the Filter window on the right panel. To access the Filter dialog, change the tab in the right panel to the filter tab. This window shows a slider or a set of checkboxes for each column in the currently active dataset (in fact, not all columns in the current dataset may be represented; unrepresented columns can be brought in using the Properties  icon on top of the filter window, and represented columns can be unrepresented here as well). Changing any of the slider or checkbox settings will remove the affected rows from ALL datasets open in the current project.

For checkboxes, you can turn multiple options on or off simultaneously rather than one by one by selecting the appropriate checkbox labels using Left-Click , Shift-Left-Click , and Ctrl-Left-Click , and then using the Clear  icon and the Select  icon.

More complex filters can be obtained by combining either the *Data →Row Commands →Label Selected Rows* command or the *Data →Column Commands →Append Columns by Formula* command along with the filter window. These operations will add new columns to the dataset and the filter window can then be use to set ranges on these columns.

2.6 Algorithms

Several different algorithms can be run on the dataset. These include Clustering, Class Prediction, Statistical Hypothesis Testing, Feature Selection, Principal Components Analysis etc. These are all accessible from the menubar. See Clustering, Classification, and Statistical Hypothesis Testing for further details.

The set of columns which are used as input in an algorithm can be chosen using the Columns tab in the dialog box of each algorithm. Most

algorithms show progress in the progress bar at the bottom of the tool and can be stopped midway using Stop icon on the toolbar.

2.7 Data Commands

The *Data* menu features various commands which can be used to add new columns to the currently active dataset or to create new datasets themselves. These commands are described below in more detail.

2.7.1 Column Operations

Commands like Logarithm, Exponent, Absolute, Scale and Threshold are mathematical operations which take as input a specified set of columns and create new transformed columns, which can either be added to the same currently active dataset or can be formed into a new child dataset.

The *Group operation* asks for two selections: the first, a set of grouping columns, and the second, a set of data columns. The rows of the currently active dataset are grouped into categories based on their values in the grouping columns; rows in a category have identical values in ALL the grouping columns. Next, for each specified data column, values within a category are averaged and a new column is created with these averaged values; all rows in a category will have the same value in this new column. This set of new columns, one for each specified data column, can either be added to the current dataset or made into a new child dataset. Note that in addition to averaging within a category, several other functions are also available, e.g., median, min, max, standard deviation, count, standard error of mean etc.

The *Remove Columns* operation can be used to remove specified columns. As mentioned in Dataset, column removal from a dataset causes the column to be removed from parent and ancestor datasets as well.

The *Import Columns* allows new columns to be brought into the dataset from specified tab or comma separated files. Specify the name of the file. In addition, you can provide the name of a column in the file as well as a column in the dataset ot be *matched by*. These columns will be used to ensure that the imported columns are matched with the order of rows in the dataset. If no column to match by is specified, then the rows will be matched by the order of occurance.

The *Append Columns by Formula* allows new columns to be created via user defined formulae. A variety of formulae are supported and examples appear on the dialog itself.

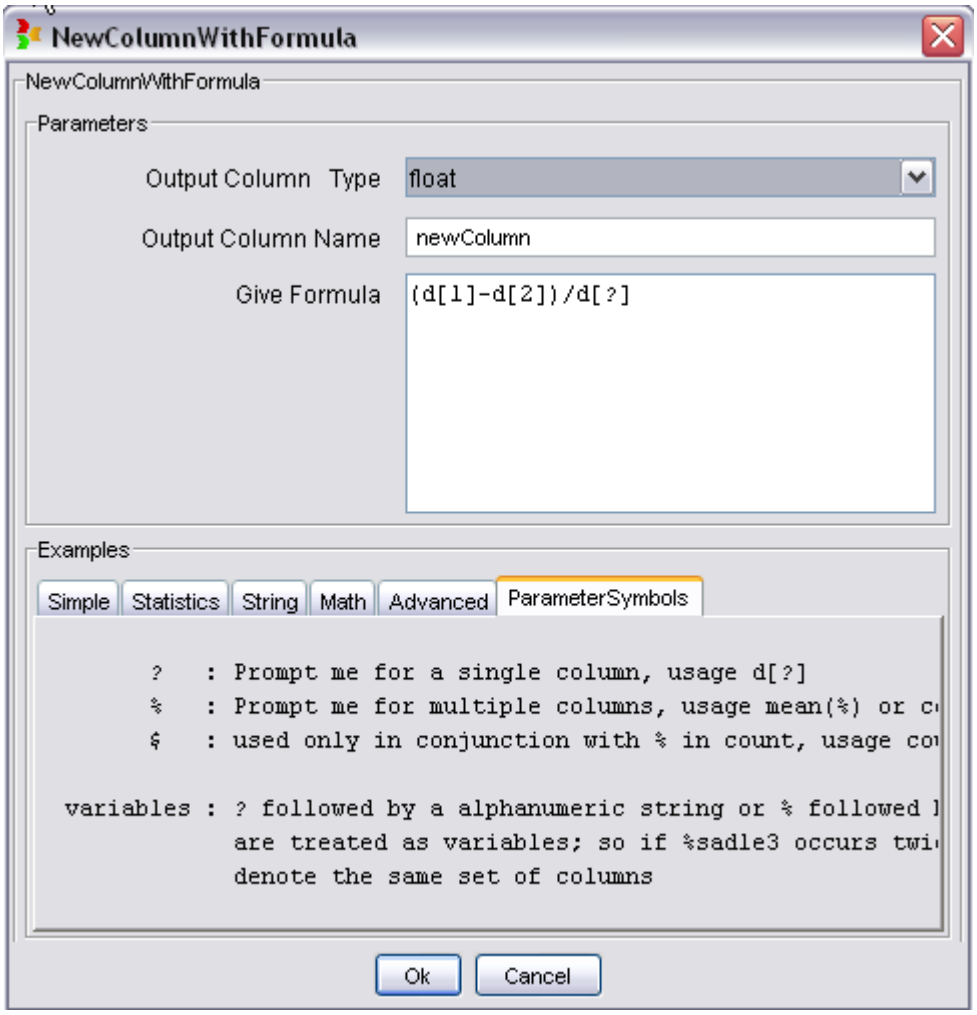


Figure 2.9: **ArrayAssist** Append Columns By Formula Dialog

2.7.2 Row Operations

The only row operation available is the *Label Selected Rows* option. This allows you to specify a label value and a particular Class Label column. It then replaces selected rows in this column by the value specified. If no column is chosed from the dron-down list, then a new column called *Label* will be appended to the dataset with the chosen label.

2.7.3 Dataset Operations

The *Create Subset* command allows you to create new child datasets by copying over subsets of rows and columns. The *Create Subset from Selection* option will take the current row and column selection in the presently active dataset and create a new child dataset comprising of only these rows and columns. The *Create Subset by Removing Selected Rows* option will take the currently active dataset and create a new child dataset comprising only unselected rows and ALL columns. The *Create SUbset by Removing Rows with Missing Values* option will take the currently active dataset and create a new child dataset comprising only rows which have no missing values and ALL columns.

The *Transpose* dataset command will create a new view in which rows of the currently active dataset become the columns and vice versa. Remember to mark an Identifier column in the currently active dataset using *Data →Data Properties* and then editing the Column Mark for the appropriate column to become Identifier. This will ensure that column headers in the new transposed view are proper. Note that this transposed view is NOT a dataset, so algorithms and graphical views cannot be derived from it. However, rows and columns in this view are indeed lassoed. To derive graphs and run algorithms from this view, use *Right-Click →Export as Text* to save this file as a txt file and then open it as a separate project using *File →Open*.

2.8 Creating Gene Lists

The Gene List window shows the gene lists that are present in the installation. Gene lists saved from any project is available across all project in **ArrayAssist**. To see the gene lists available in the tool, Right-Click on the GeneList tab in the bottem left of the tool. This will display all the gene lists available in the tool in a tree structure.

To create a gene list, select a few rows of the dataset and click on the Create gene list from selection  icon on the tool bar. This will prompt

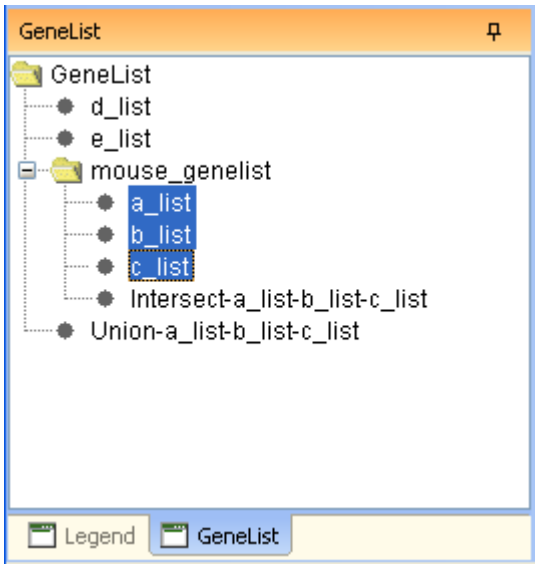


Figure 2.10: Gene Lists

a dialog where you can enter a name for the gene list and choose a mark column for the gene list from the drop-down list of the marked columns in the current dataset. This gene list will be shown in the gene list browser tree on the lower left panel of **ArrayAssist**.

Gene lists can be managed into folders and into a hierarchy tree. New folders can be created and folders can be renamed or deleted. To add a folder, to rename a folder or to delete a folder, Right-Click on a folder and choose the appropriate option. Gene lists can moved into folders by drag-and-drop into the appropriate folder.

Various operations can be performed on gene lists. These operations are all accessed by clicking on a gene list and choosing an appropriate action from the Right-Click Drop-Down-List menu.

Double-click on a gene list will select the corresponding gene in the current datasetbased on the identifier chosen. These genes will be lassoed in all the views of the dataset.

Intersect: If two or more gene lists are selected, intersect will create a gene list with the intersection of the selected gene lists. This gene list will have the genes common to all the selected gene lists. This gene list can be given a name and this will be shown in the gene list browser.

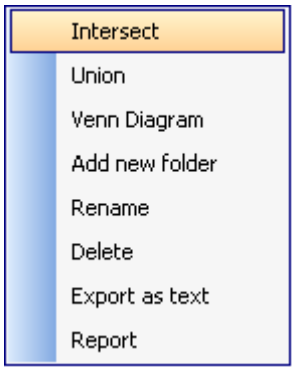


Figure 2.11: Gene Lists drop-down menu

Union: If two or more gene lists are selected, the union command will create a union of all the selected lists. This gene list will have all selected gene lists. You can give this gene list a name and this will be shown in the gene list browser.

Venn Diagram: This command will launch a venn diagram vof the two or three gene lists selected. this will create a venn diagram view showing the selected gene lists and the intersection and union of all selected lists. The numbers of genes in each sector is displayed in the venn diagram. Click on a sector will select the genes in that sector, and the selected genes will be lassoed in all the views.

Add a folder: This will add a folder to the gene list tree. You can then drag and drop gene lists into the folder.

Rename: Click on a gene list or a folder and select Rename allows you to rename the gene list or a folder.

Export as text: This will export the selected gene list as a text that contains the name of the identifier and values of the identifier for each gene.

Report: This will generate a report of the chosen gene list showing the genes in the list and a description of the gene list specifying the mark uesd to create the list.

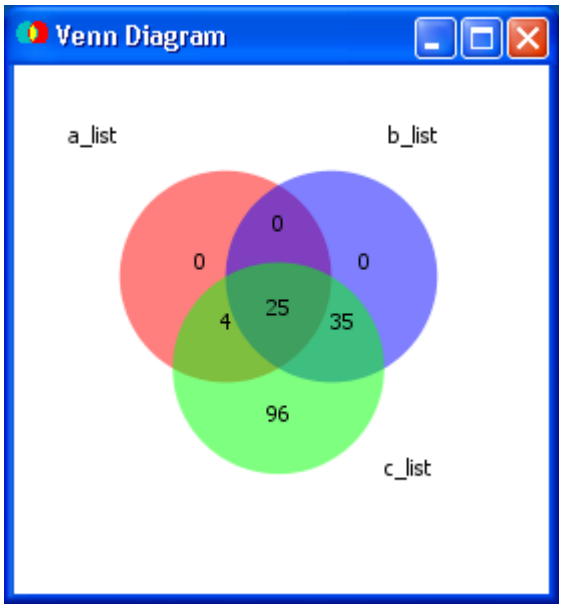


Figure 2.12: Gene Lists drop-down menu

2.9 Tiling Views

For easy simultaneous viewing of multiple windows, use the *Windows* →*Tile* option. You can set the Tiling mode to None, Vertical, Horizontal or Both. To retile views when you resize them, use Retile windows  icon.

2.10 Saving Data and Sharing Sessions

A dataset can be saved as a tab separated file using the *Right-Click* →*Export As Text* option on the corresponding spreadsheet view. The master dataset can be saved via this procedure or via *File* →*Export Data*.

In addition, an entire session comprising several open views for a dataset can be saved as a **ArrayAssist** project file .avp file; this file can then be reloaded into **ArrayAssist** to restore the entire session. To share a session with someone else, simply send them the .avp file. This session file also maintains row selections, thus allowing you to highlight some important rows to bring them into the viewer's attention.

2.11 The Log Window

Operations performed on individual projects are logged in a *Log* window associated with the project. To see the log for a particular dataset, click on Log  icon or use *View→Log*. The messages in the log window are printed at various levels of detail. The highest log level is FATAL followed by ERROR for error messages, WARN for warnings, INFO for general information and DEBUG for details.

2.12 Accessing Remote Web Sites

ArrayAssist can perform automatic, batched annotation of genes from remote web sources. See Annotation for further information.

2.13 Exporting and Printing Images and Reports

Each view can be printed as an image or as an HTML file: Right-Click on the view, use the Export As option, and choose either Image or HTML. Image format options include jpeg (compressed) and png (high resolution).

Exporting Whole Images. Exporting an image will export only the VISIBLE part of the image. Only the dendrogram view supports whole image export via the Print or Export as HTML options; you will be prompted for this. The Print option generates an HTML file with embedded images and pops up the default HTML browser to display the file. You need to explicitly print from the browser to get a hardcopy.

Finally, images can be copied directly to the clipboard and then pasted into any application like Powerpoint or Word. Right-Click on the view, use the *Right-Click Copy View* option and then paste into the target application. Further, columns in a dataset can be exported to the Windows clipboard or to another dataset as well. Select the columns in the spreadsheet and either use Right-Click followed by Copy Columns and then paste them into other applications like Excel using Ctrl-V or into other datasets using *Right-Click →Paste Columns*.

2.14 Scripting

ArrayAssist has a powerful scripting interface which allows automation of tasks within **ArrayAssist** via flexible Jython scripts. Most operations available on the **ArrayAssist** UI can be called from within a script. To run a script, go to *Tools→Script Editor*. A few sample scripts are available in the scripts subdirectory of the samples directory. For further details, refer to the Scripting chapter. In addition, R scripts can also be called via the *Tools→R Script Editor*.

2.15 Configuration

Various parameters about **ArrayAssist** are configurable from *File→Configuration*. These include algorithm parameters and various URLs.

2.16 Getting Help

Help is accessible from various places in **ArrayAssist** and always opens up in an HTML browser.

Single Button Help. Context sensitive help is accessible by pressing F1 from anywhere in the tool.

All configuration utility and dialogs have a *Help* button. Clicking on these takes you to the appropriate section of the help. All error messages with suggestions of resolution have a help button that opens the appropriate section of the online help. Additionally, hovering the cursor on an icon in any of the windows of **ArrayAssist** displays the function represented by that icon as a tool tip.

Help is accessible from the dropdown menu on the menubar. The Help menu provides access to all the documentation available in **ArrayAssist**. These are listed below:

- **Help**: This opens the Table of Contents of the on-line **ArrayAssist** user manual in a browser.
- **Documentation Index**: This provides an index of all documentation available in the tool.

- **About ArrayAssist** : This provides information on the current installation, giving the edition, version and build number.

Chapter 3

Data Visualization

3.1 View

Multiple graphical visualizations of data and analysis results are core features of **ArrayAssist** that help discover patterns in the data. All views are interactive and can be queried, linked together, configured, and printed or exported into various formats. The data views provided in **ArrayAssist** are the Spreadsheet, the Scatter Plot, the 3D Scatter Plot, the Profile Plot, the Heat Map, the Histogram, the Matrix Plot, the Summary Statistics, and the Bar Chart view. These views can be launched from the icons on the toolbar from a script, or from the **View** menu of the main menubar. All views are *lassoed*, i.e., selections on other views are propagated to these views as well.



Spreadsheet: This is a table of the raw data and it used to perform data operations.



Scatter Plot: This is 2-D plot of any two chosen columns of the active dataset.



3D Scatter Plot: This is 3-D plot of any three chosen columns of the active dataset.



Profile Plot: This is a profile plot of all rows of the dataset across chosen columns of the active dataset.



Heat Map: This is a color scaled view of the active dataset.



Histogram: This is a histogram of a selected column of the active dataset.



Matrix Plot: This is a matrix of 2-D plot of multiple chosen columns of the active dataset.



Summary Statistics: This is a descriptive statistics table of selected columns of the active dataset.



Box Whisker: This is a box whisker plot of columns in the active dataset



Bar Chart: This is a bar chart of a selected column in the dataset.

In addition to the above, there are two special views.



The Log View: Not Lassoed. Records operations performed on the current dataset.



The Lasso View: Lassoed. Shows selected rows in the current dataset.

3.1.1 View Operations

All data views and algorithm results share a common menu and a common set of operations. There are two types of views, the plot derived views, like

the Scatter Plot, the 3D Scatter plot, the Profile Plot, the Histogram, the Matrix Plot, etc.; and the table derived views like the spreadsheet, the Lasso view, the Heat Map view, the Bar Chart and various algorithm result views. Plot views share a common set of menus and operations and table views share a common set of operations and commands.

In addition, some views like the Heat Map are provided with a tool bar with icons that are specific to that particular data view. The following section below gives details of the of the common view menus and their operations. The operations specific to each data view are explained in the following sections.



Selection Mode Toggle icon: This icon appears when the active view is in the selection mode. Left-Click on this icon sets current mode to zoom mode.



Zoom Mode Toggle icon: This icon appears when the active view is in the zoom mode. Left-Click on this icon sets current mode to select mode.



Invert Selection: Inverts the current selection in the view.



Clear Selection: Clears the current selection in the view.



Reset Zoom: Resets the zoom scale to default level (i.e. shows all rows).



Print to Browser: Prints the current view to the default browser



Properties: Displays the *Properties* dialog for the current view. The Properties Dialog helps configure and control settings specific to the view. You can change the title, description and other visualization settings of the view through this dialog. The title and description added to each view is saved with .avs session file and is also exported along with the image when it is printed to HTML.

Common Operations on Plot Views

All data views and algorithm results that output a Plot share a common menu and a common set of operations. These operations are accessed from icons on the main toolbar or from Right-Click in the active canvas of the views. Views like he Scatter Plot, the 3D Scatter Plot, The profile plot, the Histogram, the Matrix Plot, etc., share a common menu and common set of operations that are detailed below.

Selection Mode: All plots are by default launched in the Selection Mode. The selection toggles with the Zoom Mode where applicable. In the selection mode, Left-Click and dragging the mouse over the view draws a selection box and selects the elements in the box. Ctrl-Left-Click and dragging the mouse over the view draws a selection box and toggles the elements in the box and add to the selection. Thus if some elements in the selection box were selected, these would become selected and if some elements in the selection box were unselected, they would be added to the already present selection.

Selection in all the views are lassoed. Thus selection on any view will be propagated to all other views.

Zoom Mode: Certain plots like the Scatter Plot and the Profile Plot allow you to zoom into specific portions of the plot. The zoom mode toggles with the selection mode. In the zoom mode, Left-Click and dragging the mouse over the view draws a zoom window with dotted lines and expands the box to the canvas of the plot.

Invert Selection: This will invert the current selection. If no elements are selected, Invert Selection will select all the elements in the current view.

Clear Selection: This will clear the current selection.

Limit to Selection: Left-Click on this check box will limit the view to the current selection. Thus only the selected elements will be shown in the current view. If there are no elements selected, there will be no elements shown in the current view. Also, when Limit to Selection is applied to the view, there will is no selection color set and the the elements will be appear in the original color in the view.

Reset Zoom: This will reset the zoom and show all elements on the canvas of the plot.

Copy View: This will copy the current view to the system clipboard. This can then be pasted into any appropriate application on the system, provided the other listens to the system clipboard.

Export Column to Dataset: Certain result views can export a column to the dataset. Whenever appropriate, the Export Column to dataset menu is activated. This will cause a column to be added to the current dataset.

Print: This will print the current active view to the system browser and will launch the default browser with the view along with the dataset name, the title of the view, with the legend and description. For certain views like the heat map, where the view is larger than the image shown, Print will pop up a dialog asking if you want to print the complete image. If you choose to print the complete image, the whole image will be printed to the default browser.

Export As: This will export the current view an Image, a HTML or the values as a text if appropriate.

- Export as Image: This will pop-up a dialog to export the view as an image. This functionality allows the user to export very high quality image. You can specify any size of the image, as well as the resolution of the image by specifying the required dots per inch (dpi) for the image. Images can be exported in various formats. Currently supported formats include png, jpg, jpeg, bmp or tiff. Finally, images of very large size and resolution can be printed in the tiff format. Very large images will be broken down into tiles and recombined after all the images pieces are written out. This ensures that memory is but built up in writing large images. If the pieces cannot be recombined, the individual pieces are written out and reported to the user. However, tiff files

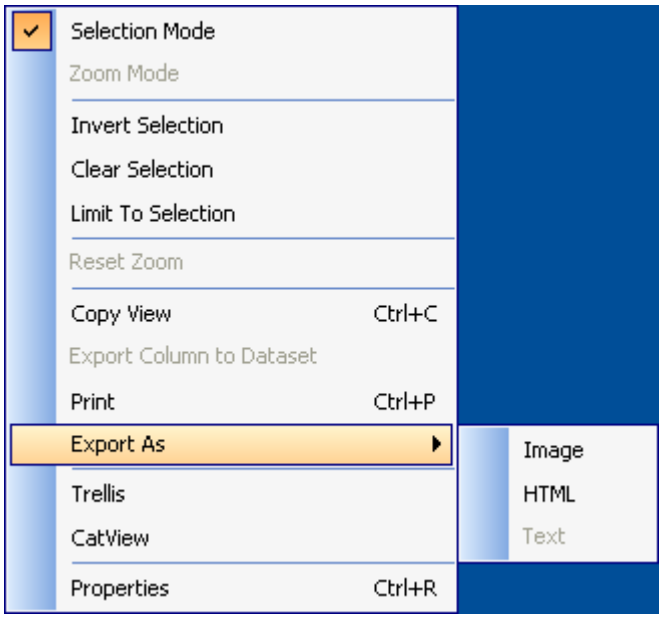


Figure 3.1: Export submenus

of any size can be recombined and written out with compression. The default dots per inch is set to 300 dpi and the default size if individual pieces for large images is set to 4 MB and tiff image without tiling enabled. These default parameters can be changed in the tools ⟶Options dialog under the **Export as Image**

Note: This functionality allows the user to create images of any size and with any resolution. This produces high-quality images and can be used for publications and posters. If you want to print vary large images or images of very high-quality the size of the image will become very large and will require huge resources. If enough resources are not available, an error and resolution dialog will pop us, saying the image is too large to be printed and suggesting you to try the tiff option, reduce the sixe of image or resolution of image, or to increase the memory avaliable to the tool by changing the -Xmx option in INSTALL_DIR/bin/packages/properties.txt file.

- Export as HTML: This will export the view as a html file. Specify the file name and the the view will ve exported as a HTML file

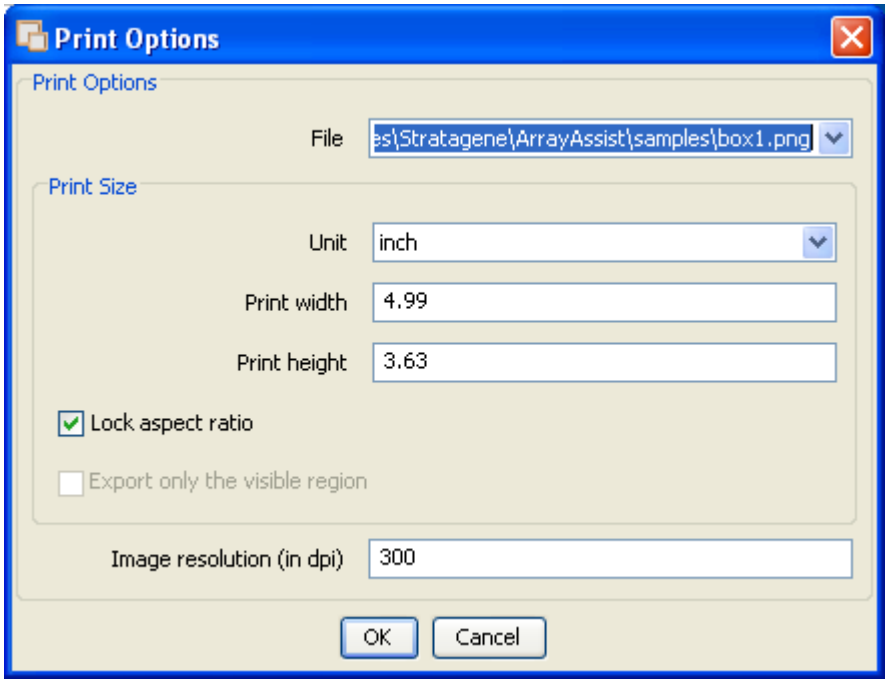


Figure 3.2: Export Image Dialog

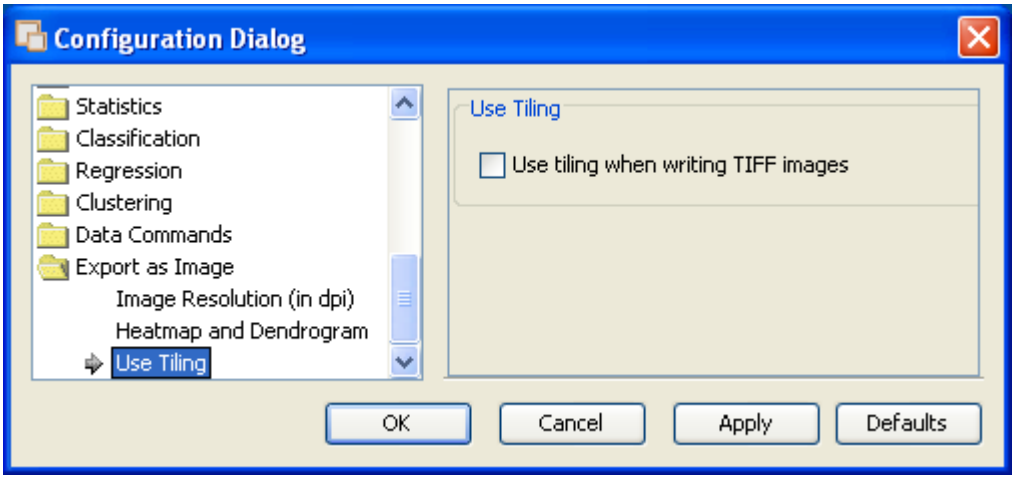


Figure 3.3: Tools ⟶Options dialog for Export as Image

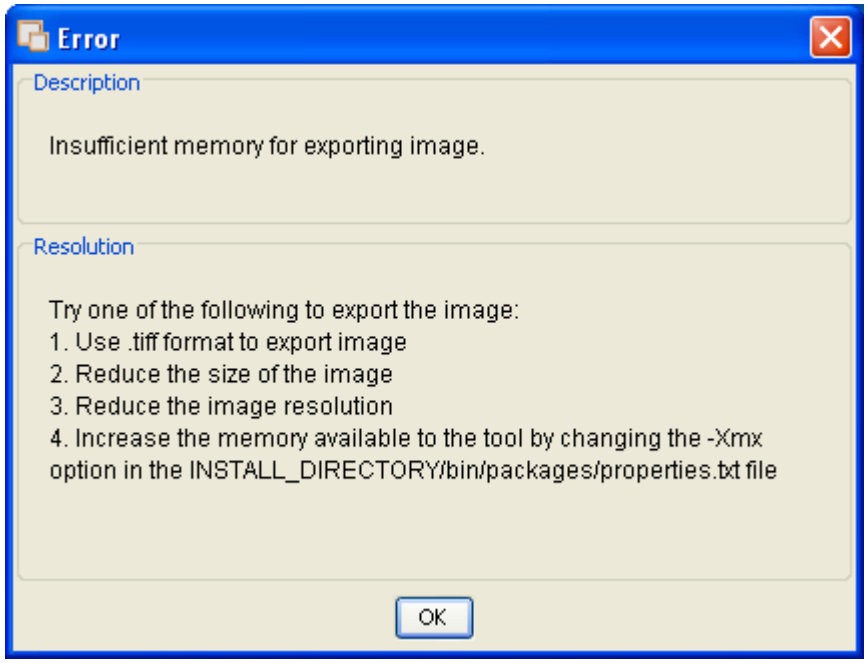

Error
Description
Insufficient memory for exporting image.
Resolution
Try one of the following to export the image:
1. Use .tiff format to export image
2. Reduce the size of the image
3. Reduce the image resolution
4. Increase the memory available to the tool by changing the -Xmx option in the INSTALL_DIRECTORY/bin/packages/properties.txt file
OK

Figure 3.4: Error Dialog on Image Export

that can be viewed in a browser and deployed on the web.

- Export as Text: Not valid for Plots and will be disabled.

Export As will pop up a file chooser for the file name and export the view to the file. Images can be exported as a jpeg, jpg or png and Export as text can be saved as txt file.

Trellis: Certain graphical views like the Scatter Plot, the Profile Plot, the Histogram, the Bar Chart, etc can be trellised on a categorical column of the dataset. This will split the dataset into different groups based upon the categories in the trellis by column and launch multiple views, one for each category in the trellis by column. By default, trellis will be launched with the trellis by column as the categorical column with the least number of categories. Trellis can be launched with a maximum of 50 categories in the trellis by column. If the dataset does not have a categorical column with less than 50 categories, an error dialog is displayed.

Cat View Certain graphical views like the Scatter Plot, the Profile Plot, the Histogram, and the Bar Chart can launch a categorical view of the parent plot based on a categorical column of the dataset. The categorical view will show the corresponding plot of only one category in a categorical column. By default, the categorical column will be the categorical column with the least number of categories in the currently active dataset. The values in the categorial column will be displayed in a drop-down list and can be changed in the categorical view. A different categorical column for the Cat View can be chosen from the right-click properties dialog of the Cat View.

Properties: This will launch the Properties dialog of the current active view. All Properties of the view can be configured from this dialog.

3.2 The Spreadsheet View

When a dataset is loaded into **ArrayAssist**, a project is created and the spreadsheet view is opened on the desktop. A spreadsheet presents a tabular view of the data. The spreadsheet view can be launched by clicking on the Spreadsheet icon  icon or from the View menu of the tool. The Spreadsheet is used to view the data

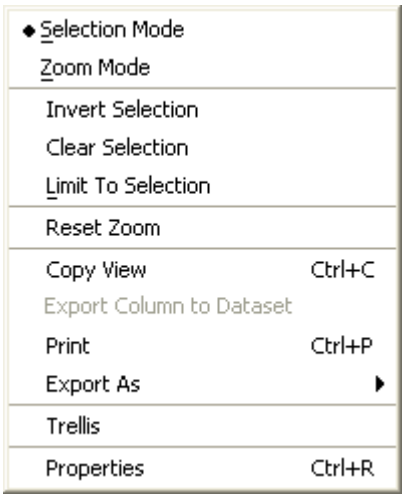


Figure 3.5: Menu accessible by Right-Click on the plot views

3.2.1 Spreadsheet Operations

Spreadsheet operations are also available by Right-Click on the canvas of the spreadsheet. Operations that are common to all views are detailed in the section Common Operations on Table Views above. In addition, some of the spreadsheet specific operations and the spreadsheet properties are explained below:

Sort: The Spreadsheet can be used to view the sorted order of data with respect to a chosen column. Sort is performed by clicking on the column header. Mouse clicks on the column header of the spreadsheet will cycle though an ascending values sort, a descending values sort and a reset sort. The column header of the sorted column will also be marked with the appropriate icon.

Thus to sort a column in the ascending, click on the column header. This will sort all rows of the spreadsheet based on the values in the chosen column. Also an icon on the column header will denote that this is the sorted column. To sort in the descending order, click again on the same column header. This will sort all the rows of the spreadsheet based on the decreasing values in this column. To reset the sort, click again on the same column. This will reset the sort and the sort icon will disappear from the column header.

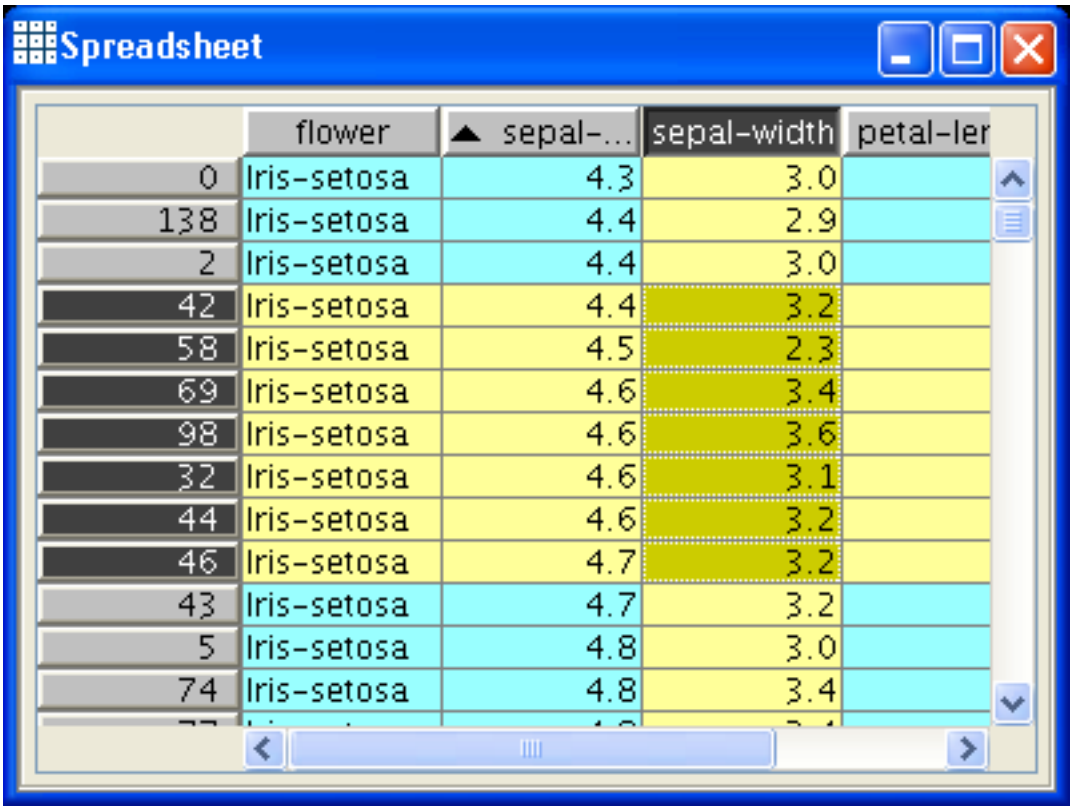


Figure 3.6: Spreadsheet

Selection: The spreadsheet can be used to select rows, columns, or any contiguous part of the dataset. The selected elements can be used to create a new dataset by Left-Click on Create dataset from Selection  icon.

Row Selection: Rows are selected by Left-Click on the row headers and dragging along the rows. Ctrl-Left-Click selects subsequent items and Shift-Left-Click selects a consecutive set of items. The selected rows will be shown in the lasso window and will be highlighted in all other views.

Column Selection: Columns can be selected by Left-Click in the column of interest. Ctrl-Left-Click selects subsequent columns and Shift-Left-Click consecutive set of columns. The current column selection on the spreadsheet usually determines the default set of selected columns used when launching any new view, executing commands or running algorithm. The selected columns will be lassoed in all relevant views and will be show selected in the lasso view.

Trellis: The spreadsheet can be trellised based on a trellis column. To trellis the spreadsheet, click on Trellis on the Right-Click menu or click Trellis from the View menu. This will launch multiple spreadsheets in the same view based on the trellis column. By default the trellis will be launched with the categorical column with the least number of categories in the current dataset. You can change the trellis column by the properties of the trellis view.

3.2.2 Spreadsheet Properties

The Spreadsheet Properties Dialog is accessible from Properties  icon on the main toolbar or by Right-Click on the spreadsheet and choosing **Properties** from the menu. The spreadsheet view can be customized and configured from the spreadsheet properties.

Rendering: The rendering tab of the spreadsheet dialog allows you to configure and customize the fonts and colors that appear in the spreadsheet view.

Special Colors: All the colors in the Table can be modified and configured. You can change the Selection color, the Double Selection color, Missing Value cell color and the Background color in the table view. To change the default colors in the view, Right-Click on

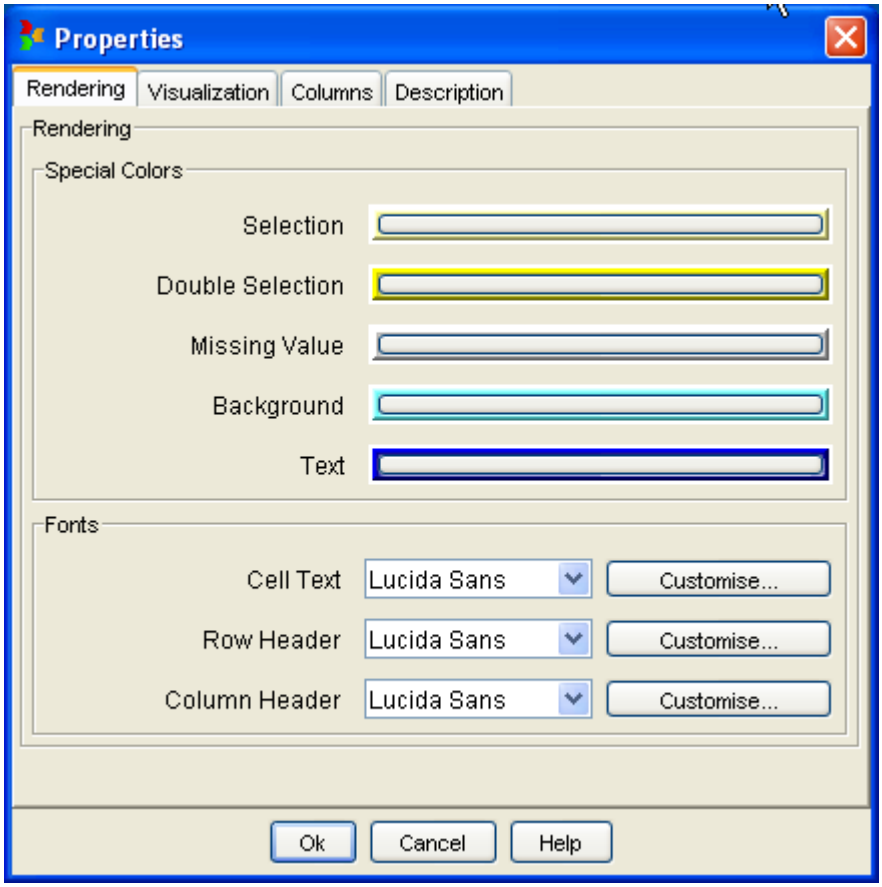


Figure 3.7: Spreadsheet Properties Dialog

the view and open the Properties dialog. Click on the Rendering tab of the properties dialog. To change a color, click on the appropriate color bar. This will pop-up a Color Chooser. Select the desired color and click OK. This will change the corresponding color in the Table.

Fonts: Fonts can be that occur in the table can be formatted and configured. You can set the fonts for Cell text, row Header and Column Header. To change the font in the view, Right-Click on the view and open the Properties dialog. Click on the Rendering tab of the Properties dialog. To change a Font, click on the appropriate drop-down box and choose the required font. To customise the font, click on the customise button. This will pop-up a dialog where you can set the font size and choose the font type as bold or italic.

Visualization: The display precision of decimal values in columns, the row height and the missing value text, and the facility to enable and disable sort are configured and customized by options in this tab.

Visualization: The visualization of the display precision of the numeric data in the table, the table cell size and the text for missing value can be configured. To change these, Right-Click on the table view and open the Properties dialog. Click on the visualization tab. This will open the Visualization panel.

To change the numeric precision. Click on the drop-down box and choose the desired precision. For decimal data columns you can choose between full precision and one to for decimal places, or representation in scientific notation. By default, full precision is displayed.

You can set the row height of the table, by entering a integer value in the text box and pressing Enter. This will change the row height in the table. By default the row height is set to 16.

You can enter any a text to show missing values. All missing values in the table will be represented by the entered value and missing values can be easily identified. By default all the missing value text is set to an empty string.

You can also enable and disable sorting on any column of the table by checking or unchecking the check box provided. By default, sort is enabled in the table. To sort the table on any column, click on the column header. This will sort the all rows of the table based on the

values in the sort column. This will also mark the sorted column with an icon to denote the sorted column. The first click on the column header will sort the column in the ascending order, the second click on the column header will sort the column in the descending order, and clicking the sorted column the third time would reset the sort.

Columns: The order of the columns in the spreadsheet can be changed by changing the order in the Columns tab in the Properties Dialog.

The columns for visualization and the order in which the columns are visualized can be chosen and configured for the column selector. Right-Click on the view and open the properties dialog. Click on the columns tab. This will open the column selector panel. The column selector panel shows the Available items on the left-side list box and the Selected items on the right-hand list box. The items in the right-hand list box are the columns that are displayed in the view in the exact order in which they appear.

To move a columns from the Available list box to the Selected list box, highlight the required items in the Available items list box and click on the right arrow in between the list boxes. This will move the highlighted columns from the Available items list box to the bottom of the Selected items list box. To move columns from the Selected items to the Available items, highlight the required items on the Selected items list box and click on the left arrow. This will move the highlight columns from the Selected items list box to the Available items list box in the exact position or order in which the column appears in the dataset.

You can also change the column ordering on the view by highlighting items in the Selected items list box and clicking on the up or down arrows. If multiple items are highlighted, the first click will consolidate the highlighted items (bring all the highlighted items together) with the first item in the specified direction. Subsequent clicks on the up or down arrow will move the highlighted items as a block in the specified direction, one step at a time until it reaches its limit. If only one item or contiguous items are highlighted in the Selected items list box, then these will be moved in the specified direction, one step at a time until it reaches its limit. To reset the order of the columns in the order in which they appear in the dataset, click on the reset icon next to the Selected items list box. This will reset the columns in the view in the way the columns appear in the view.

To highlight items, Left-Click on the required item. To highlight multiple items in any of the list boxes, Left-Click and Shift-Left-Click will highlight all contiguous items, and Left-Click and Ctrl-Left-Click will add that item to the highlight elements.

The lower portion of the Columns panel provides a utility to highlight items in the Column Selector. You can either match by Name or by Experimental Factor (if specified). To match by Name, select Match By Name from the drop down list, enter a string in the Name text box and hit Enter. This will do a substring match with the Available List and the Selected list and highlight the matches. To match by Experiment Grouping, the Experiment Grouping information must be provided in the dataset. If this is available, the Experiment Grouping drop down will show the factors. The groups in each factor will be show in the Groups list box. Selecting specific Groups from the text box will highlight the corresponding items in the Available items and Selected items box above. These can be moved as explained above. By default, the match By Name is used.

Description: The title for the view and description or annotation for the view can be configured and modified from the description tab on the properties dialog. Right-Click on the view and open the Properties dialog. Click on the Description tab. This will show the Description dialog with the current Title and Description. The title entered here appears on the title bar of the particular view and the description if any will appear in the Legend window situated in the bottom of panel on the right. These can be changed changing the text in the corresponding text boxes and clicking OK. By default, if the view is derived from running an algorithm, the description will contain the algorithm and the parameters used.

3.3 The Scatter Plot

The Scatter Plot is launched by Scatter Plot  icon on the toolbar or from View menu on the main menu bar. The Scatter Plot shows a 2-D scatter of points. The rows of the dataset are points on the scatter and the columns of the dataset are the axes. If columns are selected in the spreadsheet, the Scatter Plots is launched with two of the selected columns as the axes. If no column is selected, the Scatter Plot is launched with the first two data columns. The axes of the Scatter Plot can be changed to show any two

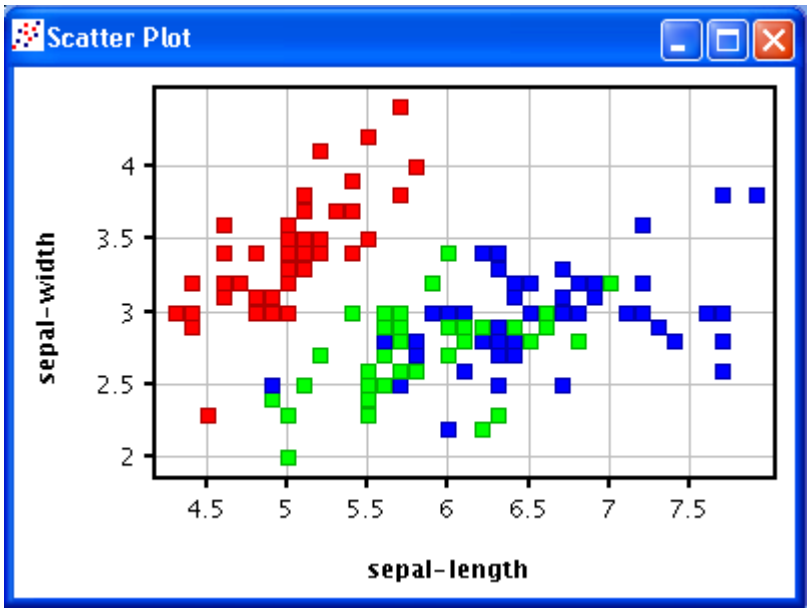


Figure 3.8: Scatter Plot

columns of the dataset from the drop down box of **X-Axis** and **Y-Axis** in the Scatter Plot.

The Scatter Plot is a lassoed view, and supports both selection and zoom modes. Most elements of the Scatter Plot, like color, shape, size of points etc. are configurable from the properties menu described below.

3.3.1 Scatter Plot Operations

Scatter Plot operations are accessed from the toolbar menu with Scatter Plot being the active window. These operations are also available by Right-Click on the canvas of the Scatter Plot. Operations that are common to all views are detailed in the section Common Operations on Plot Views. Scatter Plot specific operations and properties are discussed below.

Selection Mode: The Scatter Plot is launched in the selection mode by default. In selection mode, Left-Click and dragging the mouse over the Scatter Plot draws a selection box and all points within the selection box will be selected. To select additional points, Ctrl-Left-Click and drag the mouse over desired region. You can also draw and select re-

gions within arbitrary shapes using Shift-Left-Click and then dragging the mouse to get the desired shape.

Selections can be inverted by Left-Click on Invert Selection icon on the toolbar or from the pop-up menu on Right-Click inside the Scatter Plot. This selects all unselected points and unselect the selected points on the scatter plot. Left-Click Clear Selection icon or from the pop-up menu on Right-Click inside the Scatter Plot to clear all selection.

Zoom Mode: The Scatter Plot can be toggled from the Selection Mode to the Zoom Mode by Toggle icon on the toolbar. While in the zoom mode, Left-Click and dragging the mouse over the selected region draws a zoom box and will zoom into the region. Left-Click on the Reset Zoom icon to revert back to the default, showing all the points in the dataset.

The Scatter Plot can be trellised based on a trellis column. To trellis the Scatter Plot, click on Trellis on the Right-Click menu or click Trellis from the View menu. This will launch multiple Scatter Plot in the same view based on the trellis column. By default the trellis will be launched with the categorical column with the least number of categories in the current dataset. You can change the trellis column by the properties of the trellis view.

3.3.2 Scatter Plot Properties

The Scatter Plot view offers a wide variety of customization with log and linear scale, colors, shapes, sizes, drawing orders, error bars, line connections, titles and descriptions from the Properties dialog. These customizations appear in three different tabs on the Properties window, labelled Axis, Visualization, Rendering, Description.

Axis: The axes of the Scatter Plot can be set from the Properties Dialog or from the Scatter Plot itself. When the Scatter Plot is launched, it is drawn with the first two data columns in the dataset. If columns are selected in the spreadsheet, the Scatter Plot is launched with the first two selected data columns. These axes can be changed from the X-Axis and Y-Axis selector in the drop down box in this dialog or in the Scatter Plot itself.

The X-Axis and Y-Axis for the plot, Axis titles, the Minimum and Maximum limits for the plot, scale of the plot, the grid options, the

Trellis:

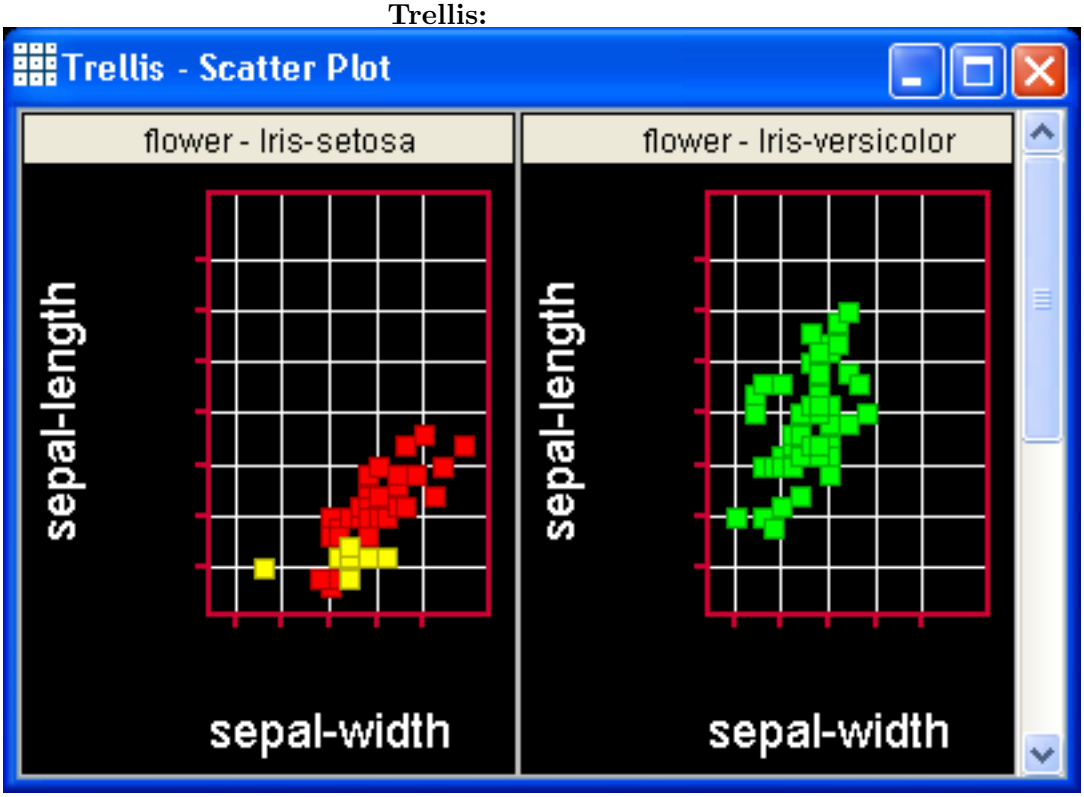


Figure 3.9: Scatter Plot Trellised

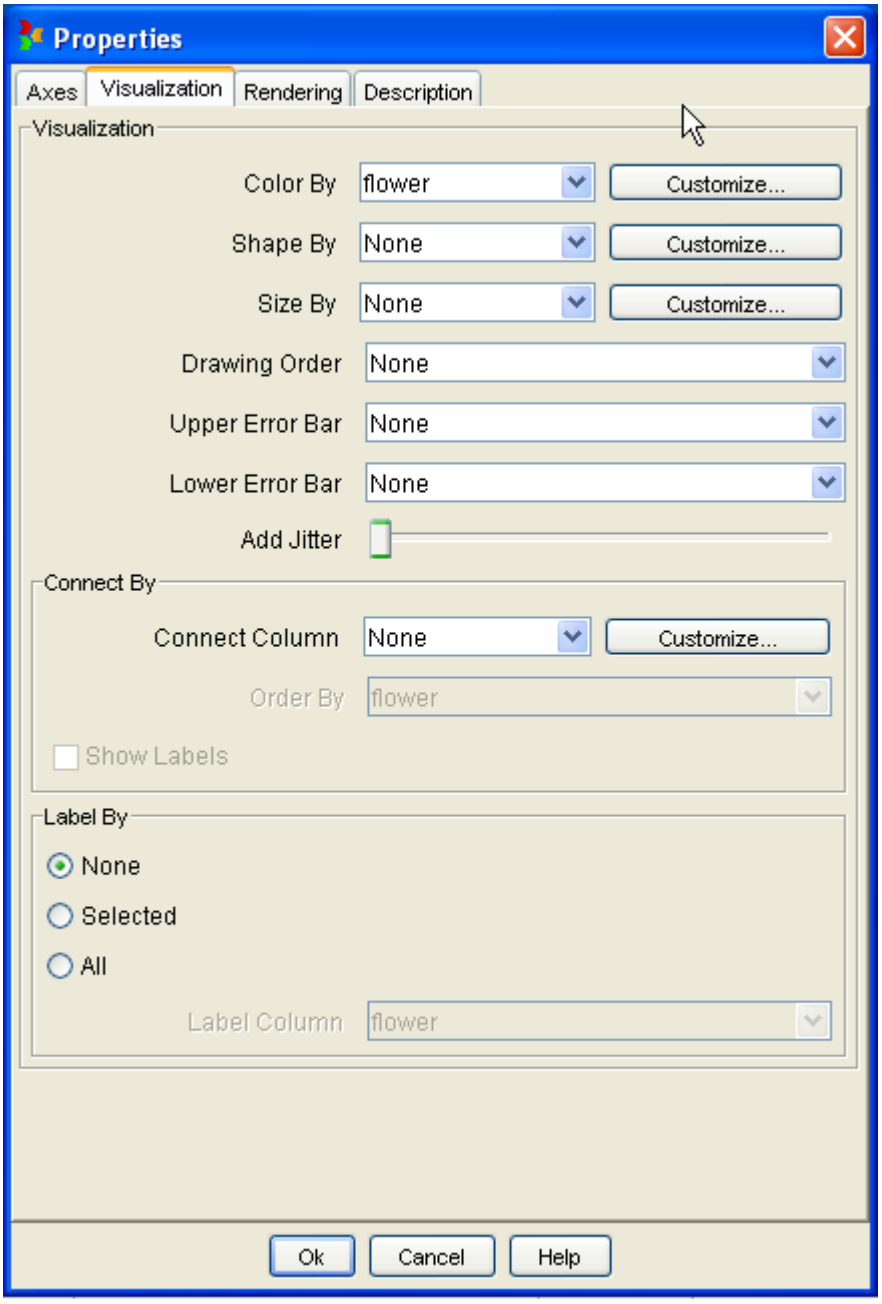


Figure 3.10: Scatter Plot Properties

label options and the number of tics on the plot can be changed and modified from the Axis tab of the Scatter Plot Properties dialog.

To change the scale of the plot to the log scale, click on the log scale option for each axis. This will provide a drop-down of the log scale options.

None: If None is chosen, the points on the chosen axis is drawn on the linear scale

Log:, If Log Scale is chosen, the points on the chosen axis is drawn on the log scale, with log of negative values if any being marked at missing values and dropped from the plot.

$(if x >), x = log(x)$

$(if x <= 0), x =$ missing value

Symmetric Log: If Symmetric Log is chosen, the points along the chosen axis are transformed such that for negative values, the log of the $1-$ absolute value is taken and plotted on the negative scale and for positive values the log of $1+$ absoulte value is taken and plotted on the positive scale.

$(if x >= 0), x = log(1 + x)$

$(if x < 0), x = -log(1 - x)$

The grids, axes labels, and the axis ticks of the plots can be configured and modified. To modify these, Right-Click on the view, and open the Properties dialog. Click on the Axis tab. This will open the axis dialog. The plot can be drawn with or without the grid lines by clicking on the show grids option. The tics and axis labels are automatically computed for the plot and show on the plot. You can show or remove the axis labels by clicking on the Show Axis Labels check box. The number of ticks on the axis are automatically computed to a show equal intervals between the minimum and maximum and displayed. You can increase the number of ticks displayed on the plot by moving the Axis Ticks slider. For continuous data columns, you can double the number of ticks shown by moving the slider to the maximum. For categorical columns, if the number of categories are less than ten, all the categories are show and moving the slider does not increase the number of tics.

Visualization: The colors, shapes and sizes of points in the Scatter Plot are configurable.

Color By: The points in the Scatter Plot can be plotted in a fixed color by clicking on the Fixed radio button. The color can also be determined by values in one of the columns by clicking the By Columns radio button and choosing the column to color by, as one of the columns in the dataset. This colors the points based on the values in the chosen columns. The color range can be modified by clicking the Customize button.

Shape By: The shape of the points on the scatter plot can be drawn with a fixed or or be based on values in any categorical column of the active dataset. To change the Shape by column, click on the drop down list provided and choose any column. Note that only categorical columns in the active dataset will be shown list. To customize the shapes, click on the customize button next to the drop down list and choose appropriate shapes.

Size By: The size of points in the scatter plot can be drawn with a fixed shape, or can be drawn based upon the values in any column of the active dataset. To change the Size By column, click on the drop down box and choose an appropriate column. This will change the plot sizes depending on the values in the particular column. You can also customize the sizes of points in the plot, by clicking on the customize button. This will pop up a dialog where the sizes can be set.

Error Bars: When visualizing profiles using the scatter plot, you can also add upper and lower error bars to each point. The length of the upper error bar for a point is determined by its value in a specified column, and likewise for the lower error bar.

If error columns are available in the current dataset,this can enable viewing Standard Error of Means via error bars on the scatter plot.

Jitter: If the points on the scatter plot are too close to each other, or are actually on top of each other, then it is not possible to view the density of points in any portion of the plot. To enable visualizing the density of plots, the jitter function is helpful. The jitter function will perturb all points on the scatter plot within a specified range, randomly, and the draw the points. the Add jitter slider specifies the range for the jitter. By default there is no jitter in the plots and the jitter range is set to zero. the jitter range can be increased by moving the slider to the right. This

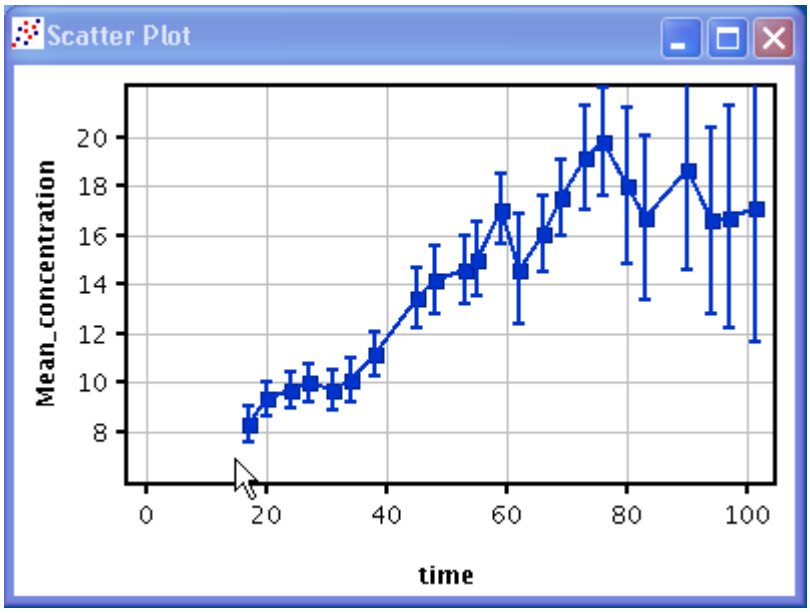


Figure 3.11: Viewing Profiles and Error Bars using Scatter Plot

will increase the jitter range and the points will now be randomly perturbed from their original values, within this range.

Connect Points: Points with the same value in a specified column can be connected together by lines in the Scatter Plot. This helps identify groups of points and also visualize profiles using the scatter plot. The column specified must be a categorical column. This column will be used to group the points together. The order in which these will be connected by lines is given by another column, namely the Order By Column. This Order By Column can be categorical or continuous.

Drawing Order: In a Scatter Plot with several points, multiple points may overlap causing only the last in the drawing order to be fully visible. You can control the drawing order of points by specifying a column name. Points will be sorted in increasing order of value in this column and drawn in that order. This column can be categorical or continuous. If this column is numeric and you wish to draw in decreasing order instead of increasing, simply scale this column by -1 using the scale operation.

Labels: You can label each point in the plot by its value in a particular column; this column can be chosen in the Label Column drop-down list. Alternatively, you can choose to label only the selected points.

Rendering: The Scatter plot allows all aspects of the view to be customizing the configured. The fonts, the colors, the offsets, etc can be configured.

Fonts: All fonts on the plot, can be formatted and configured. To change the font in the view, Right-Click on the view and open the Properties dialog. Click on the Rendering tab of the Properties dialog. To change a Font, click on the appropriate drop-down box and choose the required font. To customize the font, click on the customize button. This will pop-up a dialog where you can set the font size and choose the font type as bold or italic.

Special Colors: All the colors that occur in the plot can be modified and configured. The plot Background Color, the Axis Color, the Grid Color, the Selection Color, as well as plot specific colors can be set. To change the default colors in the view, Right-Click on the view and open the Properties dialog. Click on the Rendering tab of the Properties dialog. To change a color, click on the appropriate color bar. This will pop-up a Color Chooser. Select the desired color and click OK. This will change the corresponding color in the View.

Offsets: The left offset, right offset and the top offset and bottom offset of the plot can be modified and configured. These offsets may be need to be changed if the axis labels or axis titles are not completely visible in the plot, or if only the graph portion of the plot is required. To change the offsets, Right-Click on the view and open the Properties dialog. Click on the Rendering tab. To change plot offsets, move the corresponding slider, or enter an appropriate value in the text box provided. This will change the particular offset in the plot.

Miscellaneous: The quality of the plot can be enhanced by anti aliasing all the points in the plot. this is done to ensure better print quality. To enhance the plot quality, click on the High Quality Plot option.

Column Chooser: The column chooser can be disable and removed from the scatter plot if required. The plot area will be increased

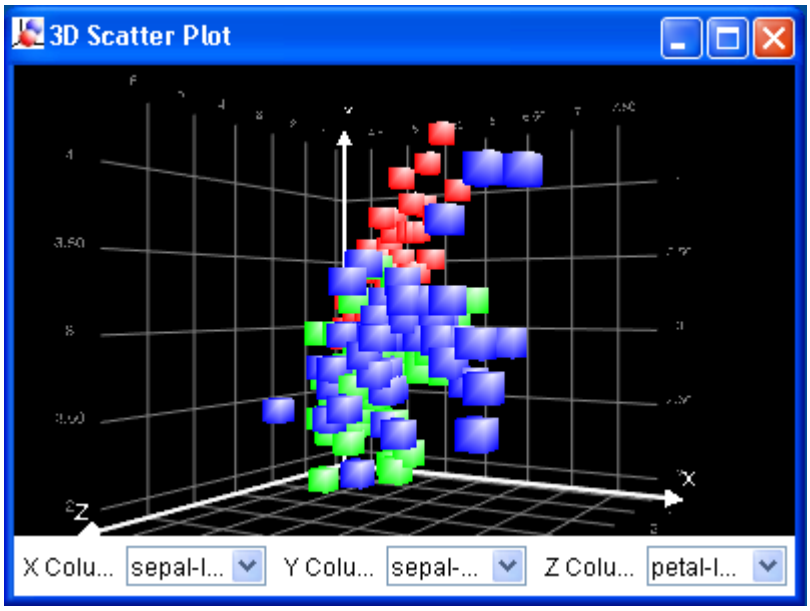


Figure 3.12: 3D Scatter Plot

and the column chooser will not be available on the scatter plot. To remove the column chooser from the plot, uncheck the Show Column Chooser option.

Description: The title for the view and description or annotation for the view can be configured and modified from the description tab on the properties dialog. Right-Click on the view and open the Properties dialog. Click on the Description tab. This will show the Description dialog with the current Title and Description. The title entered here appears on the title bar of the particular view and the description if any will appear in the Legend window situated in the bottom of panel on the right. These can be changed changing the text in the corresponding text boxes and clicking OK. By default, if the view is derived from running an algorithm, the description will contain the algorithm and the parameters used.

3.4 The 3D Scatter Plot

The 3D Scatter Plot is launched by 3D Scatter Plot  icon on the toolbar or from View menu on the main menu bar. The Scatter Plot shows a 3-D scatter of points. The rows of the dataset are points on the scatter and the columns of the dataset are the axes. If columns are selected in the spreadsheet, the 3D Scatter Plot is launched with three of the selected data columns as the axes. If no column is selected, the view is launched with the first three data columns. The axes of the Scatter Plot can be changed to show any three columns of the dataset from the drop down box of **X-Axis**, **Y-Axis** and **Z-Axis** in the 3D Scatter Plot.

The 3D Scatter Plot is a lassoed view, and supports selection as in the 2D plot. In addition, it supports zooming, rotation and translation as well. The zooming procedure for a 3D Scatter plot is very different than for the 2D Scatter plot and is described in detail below.

Note: The 3D Scatter Plot view is implemented in Java3D and some vagaries of this platform result in the 3D Scatter Pot window appearing constantly on top even when another window is moved on top. To prevent this unusual effect, the 3D window is minimised whenever any other window is moved on top of it, except when the windows are in the tiled mode. Some similar unusual effects may also be noticed when exporting the view as an image or when copying the view to the windows clipboard; in both cases, it is best to ensure that the view is not overlapping with any other views before exporting. Refer to the Frequently Asked Questions Section for more information on the known problems with 3D Scatter Plot.

3.4.1 3D Scatter Plot Operations

3D Scatter Plot operations are accessed from the toolbar menu when the plot is the active window. These operations are also available by Right-Click on the canvas of the 3D Plot. Operations that are common to all views are detailed in the section Common Operations on Plot Views. 3D Scatter Plot specific operations and properties are discussed below.

Note that to enable the Right-Click menu on the 3D Scatter Plot, you can to Right-Click in the column chooser drop down area, since Right-Click is not enabled on the canvas of the 3D Scatter plot.

Selection Mode: The 3D scatter plot is always in Selection mode. Left-Click and dragging the mouse over the Scatter Plot draws a selection box and all points within the selection box will be selected. To select

additional points, Ctrl-Left-Click and drag the mouse over desired region.

Selections can be inverted by Left-Click on Invert Selection  icon on the toolbar or from the pop-up menu on Right-Click inside the 3D Scatter Plot. This selects all unselected points and unselects the selected points on the scatter plot. Left-Click Clear Selection  icon or from the pop-up menu on Right-Click inside the 3D Scatter Plot to clear all selection.

Zooming, Rotation and Translation: To zoom into a 3D Scatter plot, press the Shift key and simultaneously hold down the middle mouse button and move the mouse upwards. To zoom out, move the mouse downwards instead. To rotate, use the left mouse button instead. To translate, use the right mouse button.

Note that rotation, zoom and translation are expensive on the 3D plot and could take time for large datasets. This time could be even larger if the points on the plots are represented by complex shapes likes spheres. Thus, it is advisable to work with just dots or tetrahedra or cubes until the image is ready for export, at which point spheres or rich spheres can be used. As an optimization, rotation, zoom and translation will convert the points to dots at the beginning of the operation and convert them back to their original shapes after the mouse is released. Thus, there may be some lag at the beginning and at the end of these operations for large datasets.

3.4.2 3D Scatter Plot Properties

The 3D Scatter Plot view allows change of axes, labelling, point shape, and point colors. These options appear in the Properties dialog and are grouped into three tabs, Axes, Visualization, Rendering and Description that are detailed below.

Axis: Axis for Plots: The axes of the 3D Scatter Plot can be set from the Properties Dialog or from the Scatter Plot itself. When the 3D Scatter Plot is launched, it is drawn with some default columns. If columns are selected in the spreadsheet, the Scatter Plot is launched with the first three selected columns. These axes can be changed from the axis selectors on the view or in this Properties Dialog itself.

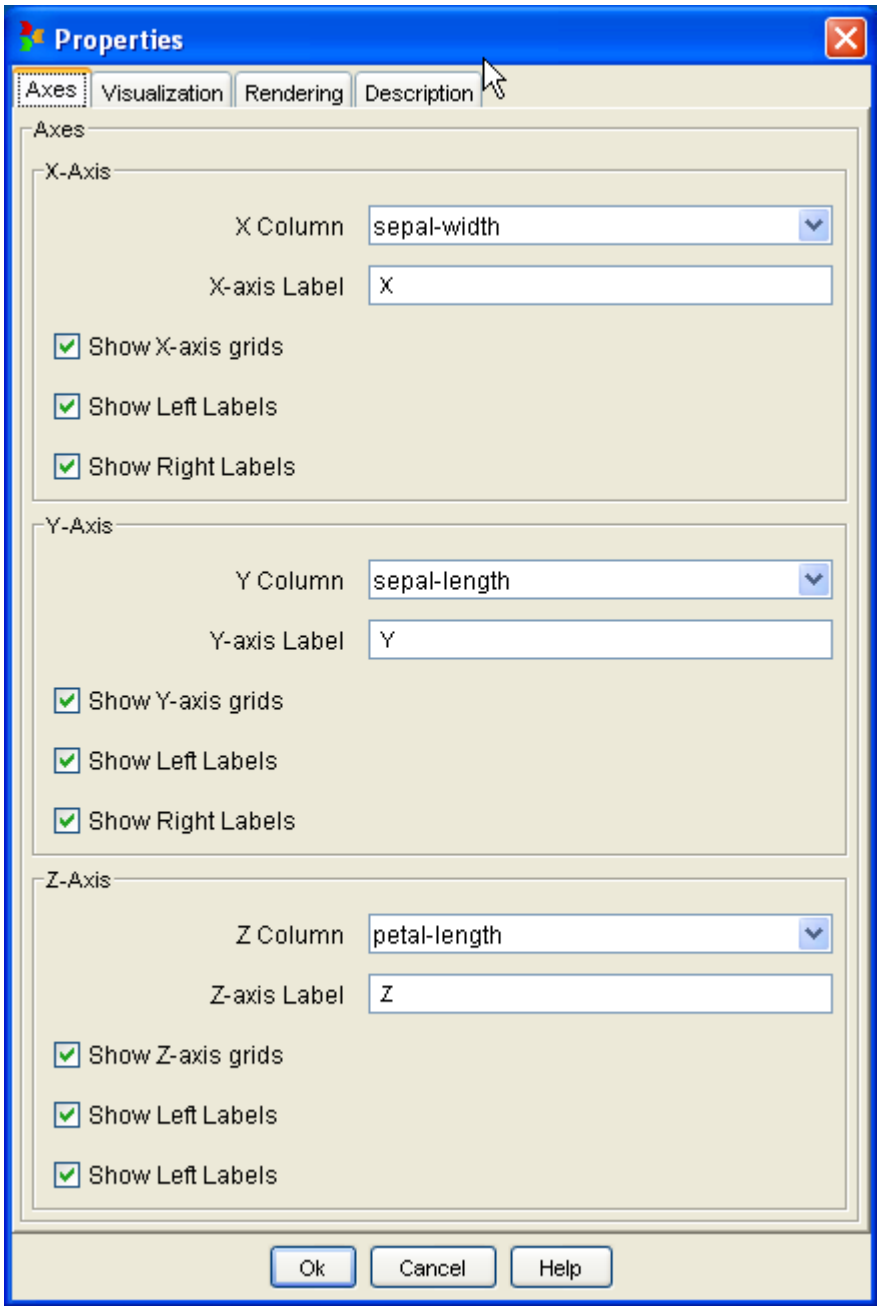


Figure 3.13: 3D Scatter Plot Properties

Axis Label: The axes are labelled by default as X, Y and Z. These default labelling can be changed by entering the new label in the Axis Label text box.

Show Grids: Points in the 3d plot are shown against a grid at the background. This grid can be disabled by unchecking the appropriate check box.

Show Labels: The value markings on each axis can also be turned on or off. Each axis has two different sets of value markings; e.g., the z-axis has one set of value markings on the xz-plane and another set of value markings on the yz-plane. These markings can be individually switched on or off using the Show Label1 and Show Label2 check boxes.

Visualization: Shape: Point shapes can be changed using the Fixed Shape drop down list of available shapes. The Dot shape will work fastest while the Rich Sphere looks best but works slowest. For large datasets (with over 2000 points), the default shape is Dot, for small datasets it is a Sphere. The recommended practice is to work with Dots, Tetrahedra or Cubes until images need to be exported.

Color By: Each point can be assigned either a fixed customizable color or a color based on its value in a specified column. Only categorical columns are allowed as choices for the 3D plot. The Customize button can be used to customize colors for both the fixed and the By-Column options.

Rendering: The colors of the 3D Scatter plot can be changed from the Rendering tab of the Properties dialog.

All the colors that occur in the plot can be modified and configured. The plot Background Color, the Axis Color, the Grid Color, the Selection Color, as well as plot specific colors can be set. To change the default colors in the view, Right-Click on the view and open the Properties dialog. Click on the Rendering tab of the Properties dialog. To change a color, click on the appropriate color bar. This will pop-up a Color Chooser. Select the desired color and click OK. This will change the corresponding color in the View.

Description: The title for the view and description or annotation for the view can be configured and modified from the description tab on the properties dialog. Right-Click on the view and open the Properties

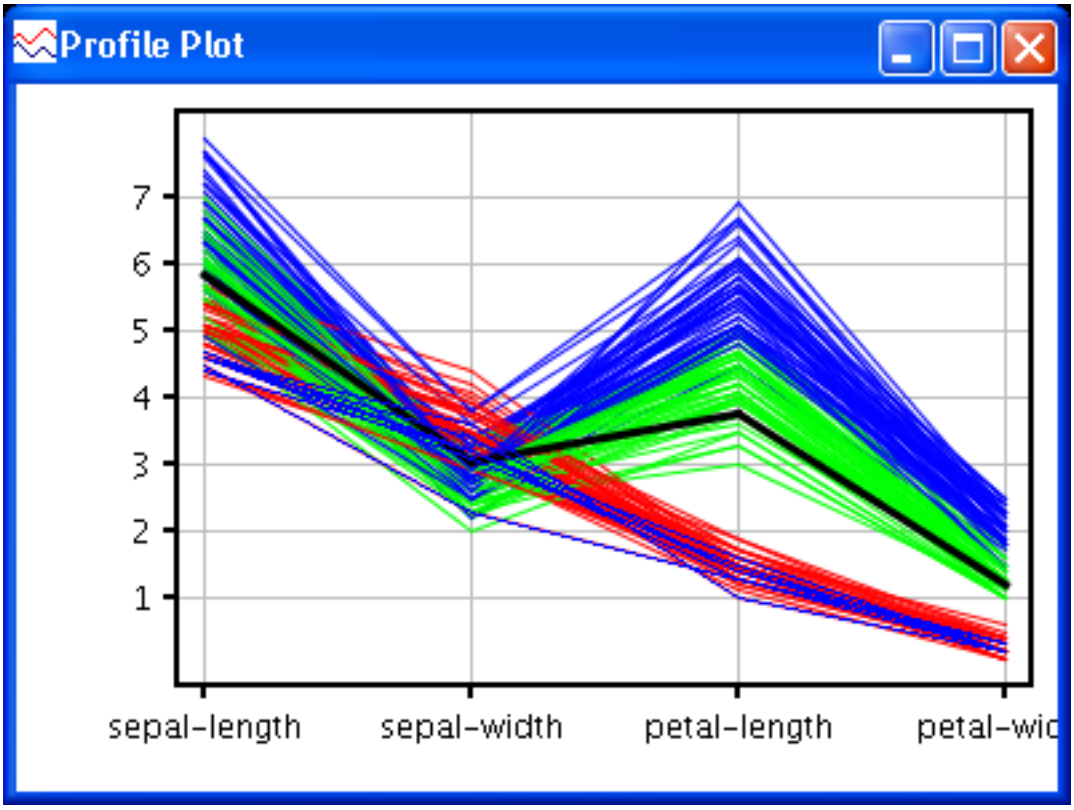


Figure 3.14: Profile Plot

dialog. Click on the Description tab. This will show the Description dialog with the current Title and Description. The title entered here appears on the title bar of the particular view and the description if any will appear in the Legend window situated in the bottom of panel on the right. These can be changed changing the text in the corresponding text boxes and clicking OK. By default, if the view is derived from running an algorithm, the description will contain the algorithm and the parameters used.

3.5 The Profile Plot View

The Profile Plot supports both the Selection Mode and the Zoom Modes. It can be launched by Left-Click on Profile Plot  icon on the main toolbar or from View menu on the main menu bar. The Profile Plot presents a view in which each row is represented as a profile over the selected columns. In addition, the mean of all these profiles is also shown on the plot in a different color. The columns represented in the plot are columns selected on the spreadsheet (if there are no columns selected then a default number of columns are sampled from the columns in the entire dataset). This column choice can be changed via Profile Plot Properties, as can the choice of colors on the plot.

3.5.1 Profile Plot Operations

The Profile Plot operations are accessed from the toolbar menu when the plot is the active window. These operations are also available by Right-Click on the canvas of the Profile Plot. Operations that are common to all views are detailed in the section Common Operations on Plot Views. Profile Plot specific operations and properties are discussed below.

Selection Mode: The Profile Plot is launched, by default, in the selection mode. While in the selection mode, Left-Click and dragging the mouse over the Profile Plot will draw a selection box and all profiles that intersect the selection box are selected. To select additional profiles, Ctrl-Left-Click and drag the mouse over desired region. Individual profiles can be selected by clicking on the profile of interest.

Zoom Mode: The Profile Plot can be toggled from the Selection Mode to the Zoom Mode by Toggle  icon on the toolbar. While in the zoom mode, Left-Click and dragging the mouse over the selected region draws a zoom box and will zoom into the region. Left-Click on the Reset Zoom  icon to revert back to the default, showing the plot for all the rows in the dataset.

Trellis: The Profile Plot can be trellised based on a trellis column. To trellis the Profile Plot, click on Trellis on the Right-Click menu or click Trellis from the View menu. This will launch multiple Profile Plot in the same view based on the trellis column. By default the trellis will be launched with the categorical column with the least number of categories in the current dataset. You can change the trellis column by the properties of the trellis view.

3.5.2 Profile Plot Properties

The following properties are configurable in the Profile Plot.

Axis: The grids, axes labels, and the axis ticks of the plots can be configured and modified. To modify these, Right-Click on the view, and open the Properties dialog. Click on the Axis tab. This will open the axis dialog. The plot can be drawn with or without the grid lines by clicking on the show grids option. The tics and axis labels are automatically computed for the plot and show on the plot. You can show or remove the axis labels by clicking on the Show Axis Labels check box. The number of ticks on the axis are automatically computed to a show equal intervals between the minimum and maximum and displayed. You can increase the number of ticks displayed on the plot by moving the Axis Ticks slider. For continuous data columns, you can double the number of ticks shown by moving the slider to the maximum. For categorical columns, if the number of categories are less than ten, all the categories are show and moving the slider does not increase the number of tics.

Visualization: The Profile Plot displays the mean profile over all rows by default. This can be hidden by unchecking the Display Mean Profile check box.

The colors of the Profile Plot can be changed from the properties dialog. The profile is drawn with a fixed color by selecting the **Fixed Color** radio button. The color can also be determined by the range of values in a chosen column by clicking the **By Column** radio button. If the color by column option is chosen, then each profile in the Profile Plot is colored based on the value of that row in that column.

Rendering: The rendering of the fonts, colors and offsets on the Profile Plot can be customized and configured.

Fonts: All fonts on the plot, can be formatted and configured. To change the font in the view, Right-Click on the view and open the Properties dialog. Click on the Rendering tab of the Properties dialog. To change a Font, click on the appropriate drop-down box and choose the required font. To customize the font, click on the customize button. This will pop-up a dialog where you can set the font size and choose the font type as bold or italic.

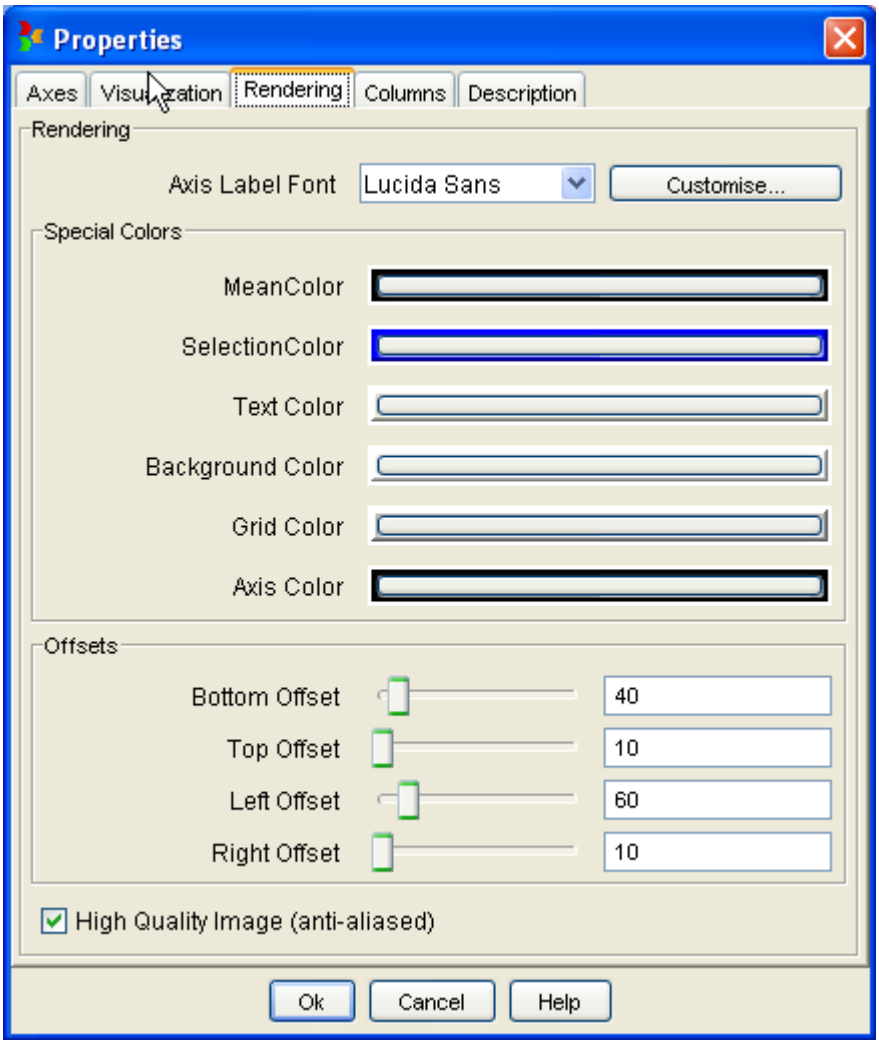


Figure 3.15: Profile Plot Properties

Special Colors: All the colors that occur in the plot can be modified and configured. The plot Background Color, the Axis Color, the Grid Color, the Selection Color, as well as plot specific colors can be set. To change the default colors in the view, Right-Click on the view and open the Properties dialog. Click on the Rendering tab of the Properties dialog. To change a color, click on the appropriate color bar. This will pop-up a Color Chooser. Select the desired color and click OK. This will change the corresponding color in the View.

Offsets: The left offset, right offset and the top offset and bottom offset of the plot can be modified and configured. These offsets may be need to be changed if the axis labels or axis titles are not completely visible in the plot, or if only the graph portion of the plot is required. To change the offsets, Right-Click on the view and open the Properties dialog. Click on the Rendering tab. To change plot offsets, move the corresponding slider, or enter an appropriate value in the text box provided. This will change the particular offset in the plot.

Quality Image: The Profile Plot image quality can be increased by checking the High-Quality anti-aliasing option. This is slow however and should be used only while printing or exporting the Profile Plot.

Column: The Profile Plot is launched with a default set of columns. The set of visible columns can be changed from the Columns tab. The columns for visualization and the order in which the columns are visualized can be chosen and configured for the column selector. Right-Click on the view and open the properties dialog. Click on the columns tab. This will open the column selector panel. The column selector panel shows the Available items on the left-side list box and the Selected items on the right-hand list box. The items in the right-hand list box are the columns that are displayed in the view in the exact order in which they appear.

To move a columns from the Available list box to the Selected list box, highlight the required items in the Available items list box and click on the right arrow in between the list boxes. This will move the highlighted columns from the Available items list box to the bottom of the Selected items list box. To move columns from the Selected items to the Available items, highlight the required items on the Selected

items list box and click on the left arrow. This will move the highlight columns from the Selected items list box to the Available items list box in the exact position or order in which the column appears in the dataset.

You can also change the column ordering on the view by highlighting items in the Selected items list box and clicking on the up or down arrows. If multiple items are highlighted, the first click will consolidate the highlighted items (bring all the highlighted items together) with the first item in the specified direction. Subsequent clicks on the up or down arrow will move the highlighted items as a block in the specified direction, one step at a time until it reaches its limit. If only one item or contiguous items are highlighted in the Selected items list box, then these will be moved in the specified direction, one step at a time until it reaches its limit. To reset the order of the columns in the order in which they appear in the dataset, click on the reset icon next to the Selected items list box. This will reset the columns in the view in the way the columns appear in the view.

To highlight items, Left-Click on the required item. To highlight multiple items in any of the list boxes, Left-Click and Shift-Left-Click will highlight all contiguous items, and Left-Click and Ctrl-Left-Click will add that item to the highlight elements.

The lower portion of the Columns panel provides a utility to highlight items in the Column Selector. You can either match by Name or by Experimental Factor (if specified). To match by Name, select Match By Name from the drop down list, enter a string in the Name text box and hit Enter. This will do a substring match with the Available List and the Selected list and highlight the matches. To match by Experiment Grouping, the Experiment Grouping information must be provided in the dataset. If this is available, the Experiment Grouping drop down will show the factors. The groups in each factor will be show in the Groups list box. Selecting specific Groups from the text box will highlight the corresponding items in the Available items and Selected items box above. These can be moved as explained above. By default, the match By Name is used.

Description: The title for the view and description or annotation for the view can be configured and modified from the description tab on the properties dialog. Right-Click on the view and open the Properties dialog. Click on the Description tab. This will show the Description

dialog with the current Title and Description. The title entered here appears on the title bar of the particular view and the description if any will appear in the Legend window situated in the bottom of panel on the right. These can be changed changing the text in the corresponding text boxes and clicking OK. By default, if the view is derived from running an algorithm, the description will contain the algorithm and the parameters used.

3.6 The Heat Map View

The Heat Map is launched by Left-Click on Heat Map  icon on the main toolbar or from View Menu on the main menu bar. The Heat Map displays numeric continuous values in the dataset as a matrix of color intensities. The expression value of each gene is mapped to a color-intensity value. The mapping of expression values to intensities is depicted by a color-bar. This provides a birds-eye view of the values in the dataset. If any columns are selected in the spreadsheet, the Heat Map is launched with the selected columns. If no columns are selected on the Spreadsheet, the Heat Map is launched with all columns in the dataset. The Heat map uses a Table view and thus allows row and column selection. The row and column selection is lassoed to all views.

3.6.1 Heat Map Operations

Heat Map operations are also available by Right-Click on the canvas of the heat map. Operations that are common to all views are detailed in the section Common Operations on Table Views above. In addition, some of the heat specific operations and the HeatMap properties are explained below:

Cell information in the Heat Map: The rows of the Heat Map correspond to the rows in the dataset and the columns in the Heat Map correspond to the columns in the dataset. If an identifier column exists in the dataset, this is used to label rows in the view. If no column is marked as an identifier, then labels will picked up from a default column in the dataset. This column choice can be customized in the Properties dialog. Mouse over any cell in the Heat Map to get the value corresponding to that cell. The mapping of values to colors can also be customized in the Properties view.

Selection Mode: The Heat Map is always in the selection mode. Select rows by clicking and dragging on the HeatMap or the row labels. It is

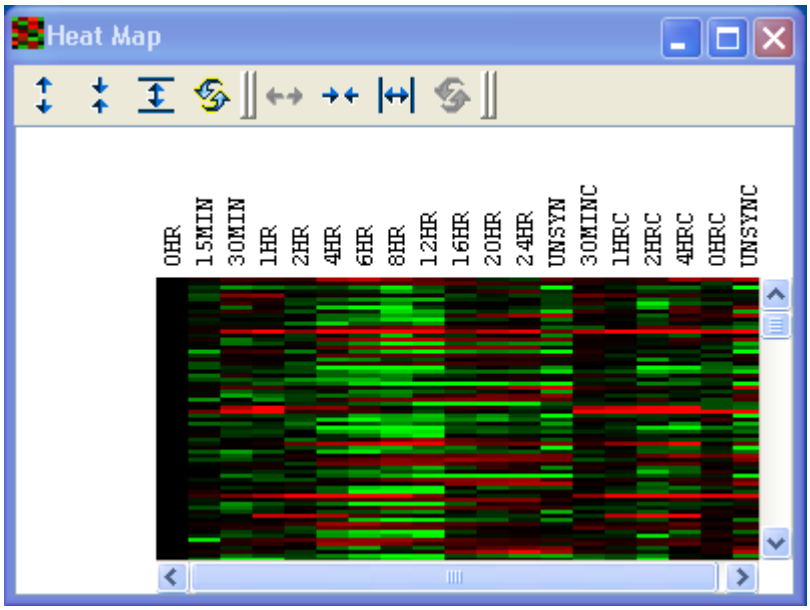


Figure 3.16: Heat Map

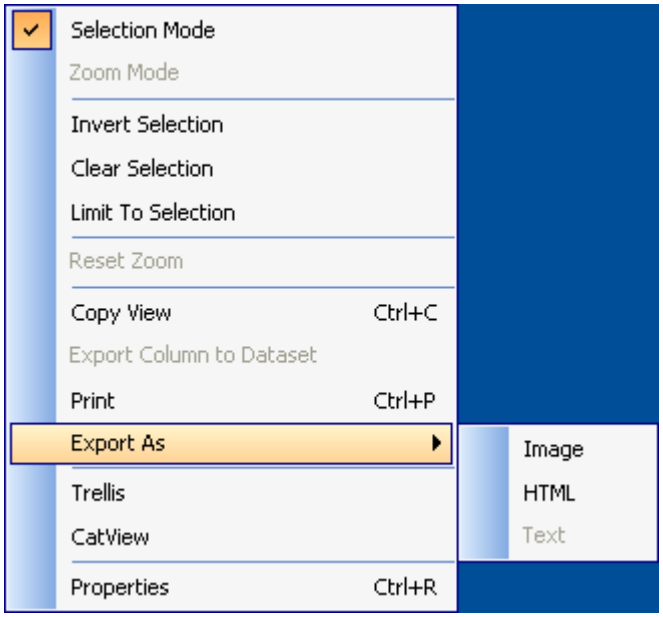


Figure 3.17: Export submenus

possible to select multiple rows and intervals using *Shift* and *Control* keys along with mouse drag. The lassoed rows are indicated in a blue overlay. Columns can also be selected in a similar manner. Both rows and columns selections are lassoed to all other views.

Export As Image: This will pop-up a dialog to export the view as an image. This functionality allows the user to export very high quality image. You can specify any size of the image, as well as the resolution of the image by specifying the required dots per inch (dpi) for the image. Images can be exported in various formats. Currently supported formats include png, jpg, jpeg, bmp or tiff. Finally, images of very large size and resolution can be printed in the tiff format. Very large images will be broken down into tiles and recombined after all the images pieces are written out. This ensures that memory is but built up in writing large images. If the pieces cannot be recombined, the individual pieces are written out and reported to the user. However, tiff files of any size can be recombined and written out with compression. The default dots per inch is set to 300 dpi and the default size if individual pieces for large images is set to 4 MB. These default parameters can be changed in the tools $\longrightarrow$Options dialog under the **Export as Image**

The user can export only the visible region or the whole image. Images of any size can be exported with high quality. If the whole image is chosen for export, however large, the image will be broken up into parts and exported. This ensures that the memory does not bloat up and that the whole high quality image will be exported. After the image is split and written out, the tool will attempt to combine all these images into a large image. In the case of png, jpg, jpeg and bmp often this will not be possible because of the size of the image and memory limitations. In such cases, the individual images will be written separately and reported. However, if a tiff image format is chosen, it will be exported as a single image however large. The final tiff image will be compressed and saved.

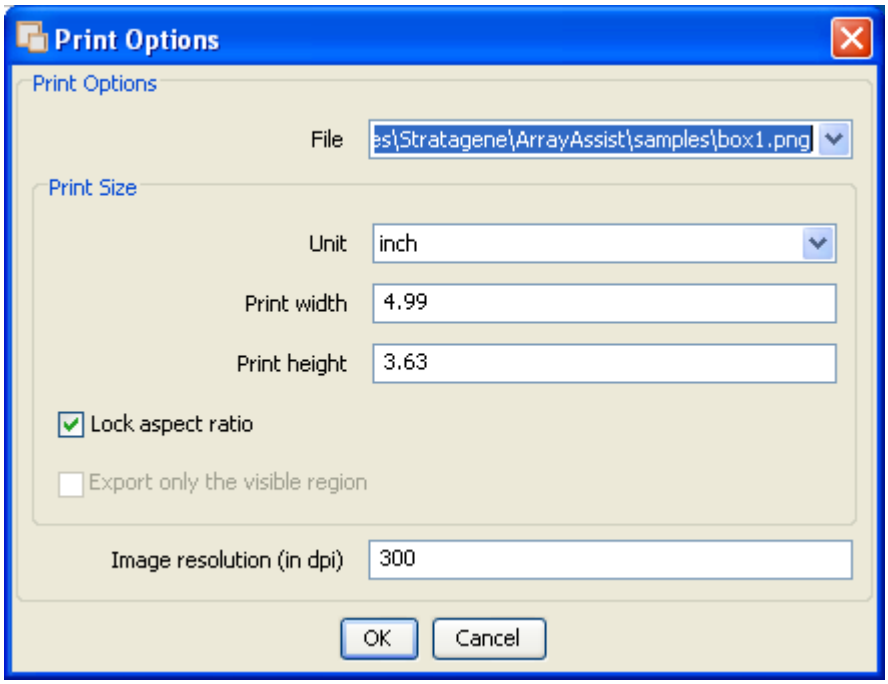


Figure 3.18: Export Image Dialog

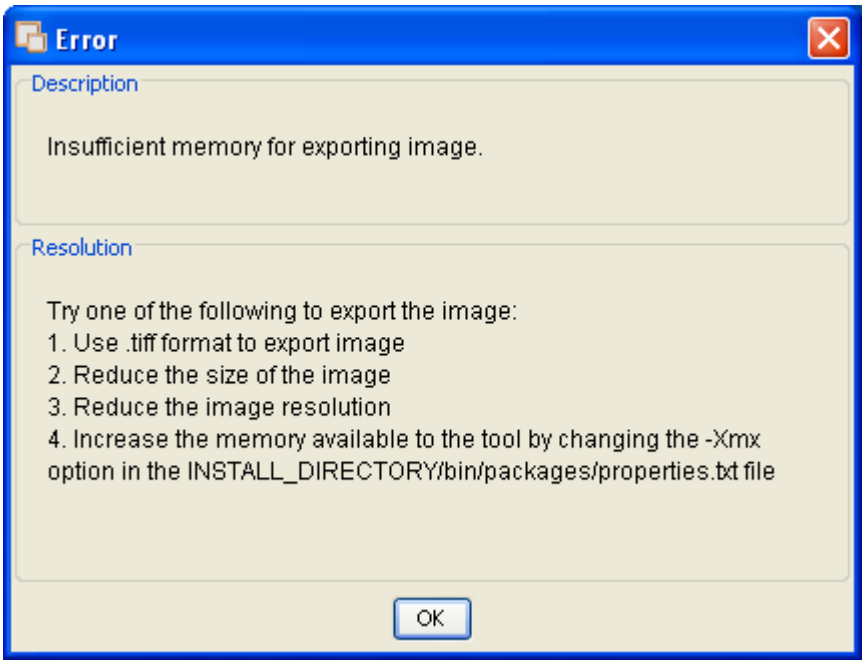

Error
Description
Insufficient memory for exporting image.
Resolution
Try one of the following to export the image:
1. Use .tiff format to export image
2. Reduce the size of the image
3. Reduce the image resolution
4. Increase the memory available to the tool by changing the -Xmx
option in the INSTALL_DIRECTORY/bin/packages/properties.txt file
OK

Figure 3.19: Error Dialog on Image Export

Note: This functionality allows the user to create images of any size and with any resolution. This produces high-quality images and can be used for publications and posters. If you want to print vary large images or images of very high-quality the size of the image will become very large and will require huge resources. If enough resources are not available, an error and resolution dialog will pop us, saying the image is too large to be printed and suggesting you to try the tiff option, reduce the sixe of image or resolution of image, or to increase the memory avaliable to the tool by changing the -Xmx option in INSTALL_DIR/bin/packages/properties.txt file.

Note: You can export the whole heat map as a single image with any size and desired resolution. To export the whole image, choose this option in the dialog. The whole image of any size can be exported as a compressed tiff file. This image can be opened on any machine with enough resources for handling large image files.



Figure 3.20: Heat Map Toolbar

Export as HTML: This will export the view as a html file. Specify the file name and the the view will ve exported as a HTML file that can be viewed in a browser and deployed on the web. If the whole image export is chosen, multiple images will be exported and can be opened in composed and open in a browser.

3.6.2 Heat Map Toolbar

The icons on the Heat Map and their operations are listed below:



Expand rows: Click to increase the row dimensions of the Heat Map. This increases the height of every row in the Heat Map. Row labels appear once the inter-row separation is large enough to accommodate label strings.



Contract rows: Click to reduce row dimensions of the Heat Map so that a larger portion of the Heat Map is visible on the screen.



Fit rows to screen: Click to scale the rows of the Heat Map to fit entirely in the window. A large image, which needs to be scrolled to view completely, fails to effectively convey the entire picture. Fitting it to the screen gives an overview of the whole dataset.



Reset rows: Click to scale the Heat Map back to default resolution showing all the row labels.
Note: Row labels are not visible when the spacing becomes too small to display labels. Zooming in or Resetting will restore these.



Expand columns: Click to scale up the Heat Map along the columns.



Contract columns: Click to reduce the scale of the Heat Map along columns. The cell width is reduced and more of the Heat Map is visible on the screen.



Fit columns to screen: Click to scale the columns of the Heat Map to fit entirely in the window. This is useful in obtaining an overview of the whole dataset. A large image, which needs to be scrolled to view completely, fails to effectively convey the entire picture. Fitting it to the screen gives a quick overview.



Reset columns: Click to scale the Heat Map back to default resolution.
Note: Column Headers are not visible when the spacing becomes too small to display labels. Zooming or Resetting will restore these.

3.6.3 Heat Map Properties

The Heat Map views supports the following configurable properties.

Visualization: Color and Saturation: The Color and Saturation Threshold of the Heat Map can be changed from the Properties Dialog. The saturation threshold can be set by the Minimum, Center and Maximum sliders or by typing a numeric value into the text box and hitting Enter. The colors of Minimum, Center and Maximum can be set from the corresponding color chooser dialog. All values above the Maximum and values below the Minimum are thresholded to Maximum and Minimum colors respectively. The chosen colors are graded and assigned to cells based on the numeric value of the cell. Values between maximum and center are assigned a graded color in between the extreme maximum and center colors, and likewise for values between minimum and center.

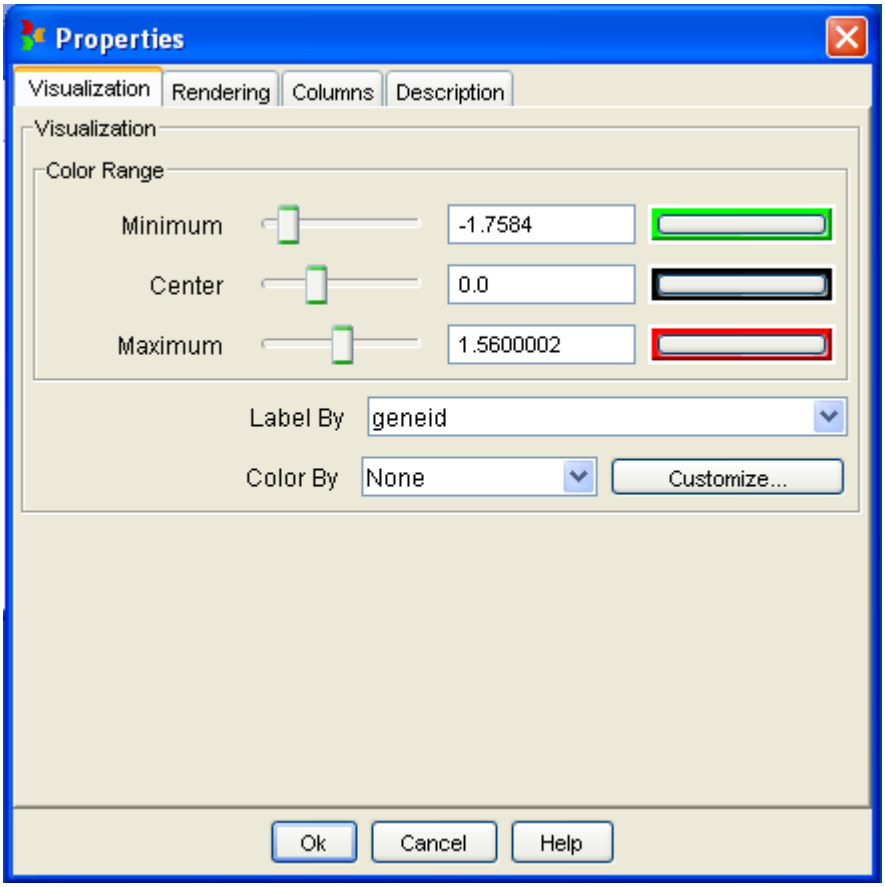


Figure 3.21: Heat Map Properties

Label Rows By: Any dataset column can be used to label the rows of the Heat Map from the **Label rows by** drop down list.

Color By: The row headers on the Heat map can be colored by categories in any categorical column of the active dataset. To color by by column, choose an appropriate column from the drop down list. Note that you can choose only categorical columns in the active dataset.

Rendering: The rendering of the Heat Map can be customized and configured from the rendering tab of the Heat map properties dialog.

To show the cell border of each cell of the Heat Map, click on the appropriate check box.

To improve the quality of the heat map by anti aliasing, click on the appropriate check box.

The row and column labels are shown along with the Heat Map. These widths allotted for these labels can be configured.

The fonts that appear in the heat map view can be changed from the drop down list provided.

Column: The Heat Map displays all columns if no columns are selected in the spreadsheet. The set of visible columns in the Heat Map can be configured from the Columns tab in properties.

The columns for visualization and the order in which the columns are visualized can be chosen and configured for the column selector. Right-Click on the view and open the properties dialog. Click on the columns tab. This will open the column selector panel. The column selector panel shows the Available items on the left-side list box and the Selected items on the right-hand list box. The items in the right-hand list box are the columns that are displayed in the view in the exact order in which they appear.

To move a columns from the Available list box to the Selected list box, highlight the required items in the Available items list box and click on the right arrow in between the list boxes. This will move the highlighted columns from the Available items list box to the bottom of the Selected items list box. To move columns from the Selected items to the Available items, highlight the required items on the Selected items list box and click on the left arrow. This will move the highlight columns from the Selected items list box to the Available items list

box in the exact position or order in which the column appears in the dataset.

You can also change the column ordering on the view by highlighting items in the Selected items list box and clicking on the up or down arrows. If multiple items are highlighted, the first click will consolidate the highlighted items (bring all the highlighted items together) with the first item in the specified direction. Subsequent clicks on the up or down arrow will move the highlighted items as a block in the specified direction, one step at a time until it reaches its limit. If only one item or contiguous items are highlighted in the Selected items list box, then these will be moved in the specified direction, one step at a time until it reaches its limit. To reset the order of the columns in the order in which they appear in the dataset, click on the reset icon next to the Selected items list box. This will reset the columns in the view in the way the columns appear in the view.

To highlight items, Left-Click on the required item. To highlight multiple items in any of the list boxes, Left-Click and Shift-Left-Click will highlight all contiguous items, and Left-Click and Ctrl-Left-Click will add that item to the highlight elements.

The lower portion of the Columns panel provides a utility to highlight items in the Column Selector. You can either match by Name or by Experimental Factor (if specified). To match by Name, select Match By Name from the drop down list, enter a string in the Name text box and hit Enter. This will do a substring match with the Available List and the Selected list and highlight the matches. To match by Experiment Grouping, the Experiment Grouping information must be provided in the dataset. If this is available, the Experiment Grouping drop down will show the factors. The groups in each factor will be show in the Groups list box. Selecting specific Groups from the text box will highlight the corresponding items in the Available items and Selected items box above. These can be moved as explained above. By default, the match By Name is used.

Description: The title for the view and description or annotation for the view can be configured and modified from the description tab on the properties dialog. Right-Click on the view and open the Properties dialog. Click on the Description tab. This will show the Description dialog with the current Title and Description. The title entered here appears on the title bar of the particular view and the description

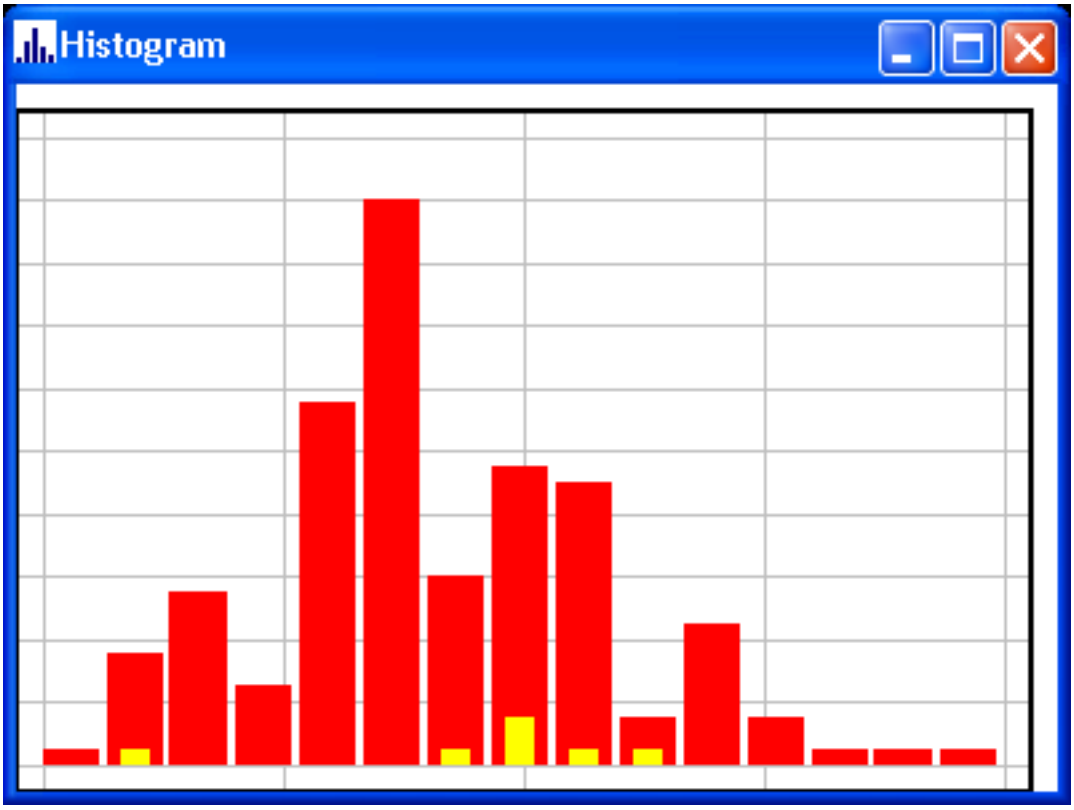


Figure 3.22: Histogram

if any will appear in the Legend window situated in the bottom of panel on the right. These can be changed changing the text in the corresponding text boxes and clicking OK. By default, if the view is derived from running an algorithm, the description will contain the algorithm and the parameters used.

3.7 The Histogram View

The Histogram is launched by Left-Click on Histogram icon on the tool bar or from View menu on the main menu bar. The Histogram presents one column (called Channel in Histogram terminology) of the dataset as a bar chart showing the frequency or number of elements in each interval of

the chosen column. This is done by binning the data in the column into equal interval bins and plotting the number of elements in each bin. If a categorical-valued column is chosen, the number of elements for each category are plotted. The frequency in each bin of the histogram is dependent upon the lower and upper limits of binning, and the size of each bin. These can be configured and changed from the **Properties** dialog. If a column is selected in the spreadsheet, the Histogram is launched with the selected column, otherwise an appropriate column is chosen automatically. The channel for the Histogram can be changed from the drop down list at the bottom of the view or from the Properties Dialog.

3.7.1 Histogram Operations

The Heat Map operations are accessed from the toolbar menu when the plot is the active window. These operations are also available by Right-Click on the canvas of the Heat Map. Operations that are common to all views are detailed in the section Common Operations on Plot Views. Heat Map specific operations and properties are discussed below.

Selection Mode: The Histogram supports only the Selection mode. Left-Click and dragging the mouse over the Histogram draws a selection box and all bars that intersect the selection box are selected and lassoed. Clicking on a bar also selects the elements in that bar. To select additional elements, Ctrl-Left-Click and drag the mouse over the desired region.

Trellis: The histogram can be trellised based on a trellis column. To trellis the histogram, click on Trellis on the Right-Click menu or click Trellis from the View menu. This will launch multiple Histograms in the same view based on the trellis column. By default the trellis will be launched with the categorical column with the least number of categories in the current dataset. You can change the trellis column by the properties of the trellis view.

3.7.2 Histogram Properties

The Histogram can be viewed with different channels, user-defined binning, different colors, and titles and descriptions from the Histogram Properties Dialog.

The Histogram Properties Dialog is accessible from Properties  icon on the main toolbar or by Right-Click on the histogram and choosing **Prop-**

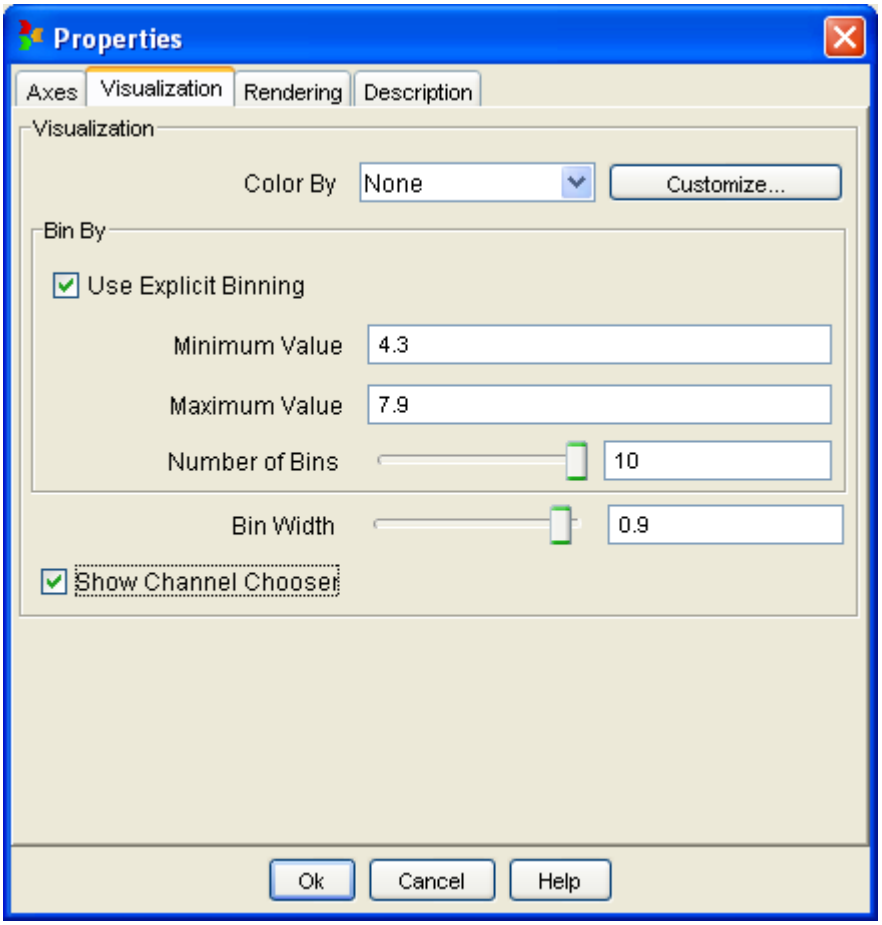


Figure 3.23: Histogram Properties

erties from the menu. The histogram view can be customized and configured from the histogram properties.

Axis: The histogram channel can be changed from the properties menu. Any column in the dataset can be selected here.

The grids, axes labels, and the axis ticks of the plots can be configured and modified. To modify these, Right-Click on the view, and open the Properties dialog. Click on the Axis tab. This will open the axis dialog. The plot can be drawn with or without the grid lines by clicking on the show grids option. The tics and axis labels are automatically computed for the plot and show on the plot. You can show or remove the axis labels by clicking on the Show Axis Labels check box. The number of ticks on the axis are automatically computed to a show equal intervals between the minimum and maximum and displayed. You can increase the number of ticks displayed on the plot by moving the Axis Ticks slider. For continuous data columns, you can double the number of ticks shown by moving the slider to the maximum. For categorical columns, if the number of categories are less than ten, all the categories are show and moving the slider does not increase the number of tics.

Visualization: Color By: You can specify a Color By column for the histogram. The Color By should be a categorical column in the active dataset. This will color each bar of the histogram with different color bars for the frequency of each category in the particular bin.

Explicit Binning: The Histogram is launched with a default set of equal interval bins for the chosen column. This default is computed by dividing the interquartile range of the column values into three bins and expanding these equal interval bins for the whole range of data in the chosen column. The Histogram view is dependent upon binning and the default number of bins may not be appropriate for the data. The data can be explicitly rebinned by checking the Use Explicit Binning check box and specifying the minimum value, the maximum value and the number of bins using the sliders. The maximum - minimum values and the number of bins can also be specified in the text box next to the sliders. Please note that if you type values into the text box, you will have to hit Enter for the values to be accepted.

Bar Width: the bar width of the histogram can be increased or decreased by moving the slider. The default is set to 0.9 times the area allocated to each histogram bar. This can be reduced if desired.

Channel chooser: The Channel Chooser on the histogram view can be disabled by unchecking the check box. This will afford a larger area to view the histogram.

Rendering: This tab provides the interface to customize and configure the fonts, the colors and the offsets of the plot.

Fonts: All fonts on the plot, can be formatted and configured. To change the font in the view, Right-Click on the view and open the Properties dialog. Click on the Rendering tab of the Properties dialog. To change a Font, click on the appropriate drop-down box and choose the required font. To customize the font, click on the customize button. This will pop-up a dialog where you can set the font size and choose the font type as bold or italic.

Special Colors: All the colors that occur in the plot can be modified and configured. The plot Background Color, the Axis Color, the Grid Color, the Selection Color, as well as plot specific colors can be set. To change the default colors in the view, Right-Click on the view and open the Properties dialog. Click on the Rendering tab of the Properties dialog. To change a color, click on the appropriate color bar. This will pop-up a Color Chooser. Select the desired color and click OK. This will change the corresponding color in the View.

Offsets: The left offset, right offset and the top offset and bottom offset of the plot can be modified and configured. These offsets may be need to be changed if the axis labels or axis titles are not completely visible in the plot, or if only the graph portion of the plot is required. To change the offsets, Right-Click on the view and open the Properties dialog. Click on the Rendering tab. To change plot offsets, move the corresponding slider, or enter an appropriate value in the text box provided. This will change the particular offset in the plot.

Description: The title for the view and description or annotation for the view can be configured and modified from the description tab on the properties dialog. Right-Click on the view and open the Properties

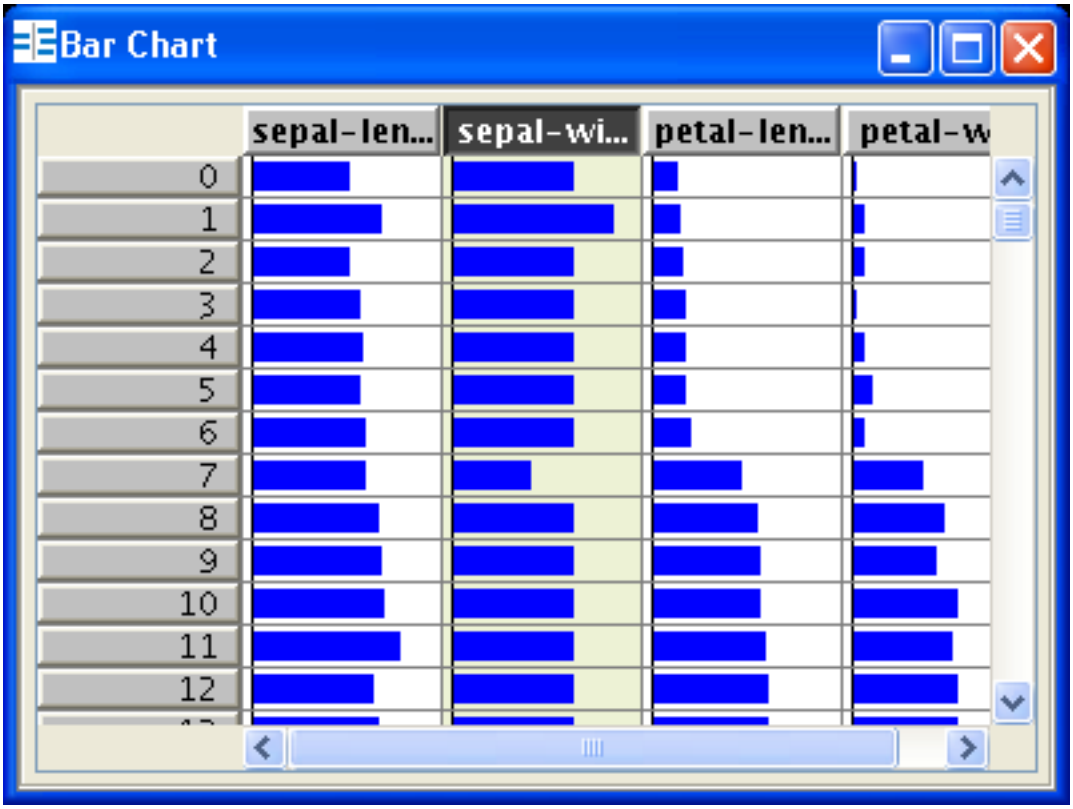


Figure 3.24: Bar Chart

dialog. Click on the Description tab. This will show the Description dialog with the current Title and Description. The title entered here appears on the title bar of the particular view and the description if any will appear in the Legend window situated in the bottom of panel on the right. These can be changed changing the text in the corresponding text boxes and clicking OK. By default, if the view is derived from running an algorithm, the description will contain the algorithm and the parameters used.

3.8 The Bar Chart

The Bar Chart is launched by Left-Click on the Bar Chart  icon on the main toolbar or from the View menu on the main menu bar. If columns are selected on any of the table views, then the Bar Chart is launched with the continuous columns in the selection. Else, by default, the Bar Chart is launched with all continuous columns in the active dataset. The Bar Chart provides a view of the range and distribution of values in the selected column. This is Bar Chart is a table view and thus all operations and that are possible on a table are possible here. The Bar Chart can be customized and configured from the **Properties** dialog accessed from the Right-Click menu on the canvas of the Chart or from the icon on the tool bar.

Note that the Bar Chart will show only the continuous columns in the current dataset.

3.8.1 Bar Chart Operations

view.BarChart.operations

The Operations on the Bar Chart is accessible from the menu on Right-Click on the canvas of the Bar Chart. Operations that are common to all views are detailed in the section Common Operations on Table Views above. In addition, some of the bar chart specific operations and the bar chart properties are explained below:

Sort: The Bar Chart can be used to view the sorted order of data with respect to a chosen column as bars. Sort is performed by clicking on the column header. Mouse clicks on the column header of the bar chart will cycle though an ascending values sort, a descending values sort and a reset sort. The column header of the sorted column will also be marked with the appropriate icon.

Thus to sort a column in the ascending, click on the column header. This will sort all rows of the bar chart based on the values in the chosen column. Also an icon on the column header will denote that this is the sorted column. To sort in the descending order, click again on the same column header. This will sort all the rows of the bar chart based on the decreasing values in this column. To reset the sort, click again on the same column. This will reset the sort and the sort icon will disappear from the column header.

Selection: The bar chart can be used to select rows, columns, or any contiguous part of the dataset. The selected elements can be used to

create a subset dataset by Left-Click on Create dataset from Selection  icon.

Row Selection: Rows are selected by Left-Click on the row headers and dragging along the rows. Ctrl-Left-Click selects subsequent items and Shift-Left-Click selects a consecutive set of items. The selected rows will be shown in the lasso window and will be highlighted in all other views.

Column Selection: Columns can be selected by Left-Click in the column of interest. Ctrl-Left-Click selects subsequent columns and Shift-Left-Click consecutive set of columns. The current column selection on the bar chart usually determines the default set of selected columns used when launching any new view, executing commands or running algorithm. The selected columns will be lassoed in all relevant views and will be show selected in the lasso view.

Trellis: The bar chart can be trellised based on a trellis column. To trellis the bar chart, click on Trellis on the Right-Click menu or click Trellis from the View menu. This will launch multiple bar chart in the same view based on the trellis column. By default the trellis will be launched with the categorical column with the least number of categories in the current dataset. You can change the trellis column by the properties of the trellis view.

3.8.2 Bar Chart Properties

The Bar Chart Properties Dialog is accessible from Properties  icon on the main toolbar or by Right-Click on the bar chart and choosing **Properties** from the menu. The bar chart view can be customized and configured from the bar chart properties.

Rendering: The rendering tab of the bar chart dialog allows you to configure and customize the fonts and colors that appear in the bar chart view.

Special Colors: All the colors in the Table can be modified and configured. You can change the Selection color, the Double Selection color, Missing Value cell color and the Background color in the table view. To change the default colors in the view, Right-Click on the view and open the Properties dialog. Click on the Rendering

tab of the properties dialog. To change a color, click on the appropriate color bar. This will pop-up a Color Chooser. Select the desired color and click OK. This will change the corresponding color in the Table.

Fonts: Fonts can be that occur in the table can be formatted and configured. You can set the fonts for Cell text, row Header and Column Header. To change the font in the view, Right-Click on the view and open the Properties dialog. Click on the Rendering tab of the Properties dialog. To change a Font, click on the appropriate drop-down box and choose the required font. To customise the font, click on the customise button. This will pop-up a dialog where you can set the font size and choose the font type as bold or italic.

Visualization: The display precision of decimal values in columns, the row height and the missing value text, and the facility to enable and disable sort are configured and customized by options in this tab.

Visualization: The visualization of the display precision of the numeric data in the table, the table cell size and the text for missing value can be configured. To change these, Right-Click on the table view and open the Properties dialog. Click on the visualization tab. This will open the Visualization panel.

To change the numeric precision. Click on the drop-down box and choose the desired precision. For decimal data columns you can choose between full precision and one to for decimal places, or representation in scientific notation. By default, full precision is displayed.

You can set the row height of the table, by entering a integer value in the text box and pressing Enter. This will change the row height in the table. By default the row height is set to 16.

You can enter any a text to show missing values. All missing values in the table will be represented by the entered value and missing values can be easily identified. By default all the missing value text is set to an empty string.

You can also enable and disable sorting on any column of the table by checking or unchecking the check box provided. By default, sort is enabled in the table. To sort the table on any column, click on the column header. This will sort the all rows of the table based on the values in the sort column. This will also mark the sorted column with

an icon to denote the sorted column. The first click on the column header will sort the column in the ascending order, the second click on the column header will sort the column in the descending order, and clicking the sorted column the third time would reset the sort.

Columns: The order of the columns in the bar chart can be changed by changing the order in the Columns tab in the Properties Dialog.

The columns for visualization and the order in which the columns are visualized can be chosen and configured for the column selector. Right-Click on the view and open the properties dialog. Click on the columns tab. This will open the column selector panel. The column selector panel shows the Available items on the left-side list box and the Selected items on the right-hand list box. The items in the right-hand list box are the columns that are displayed in the view in the exact order in which they appear.

To move a columns from the Available list box to the Selected list box, highlight the required items in the Available items list box and click on the right arrow in between the list boxes. This will move the highlighted columns from the Available items list box to the bottom of the Selected items list box. To move columns from the Selected items to the Available items, highlight the required items on the Selected items list box and click on the left arrow. This will move the highlight columns from the Selected items list box to the Available items list box in the exact position or order in which the column appears in the dataset.

You can also change the column ordering on the view by highlighting items in the Selected items list box and clicking on the up or down arrows. If multiple items are highlighted, the first click will consolidate the highlighted items (bring all the highlighted items together) with the first item in the specified direction. Subsequent clicks on the up or down arrow will move the highlighted items as a block in the specified direction, one step at a time until it reaches its limit. If only one item or contiguous items are highlighted in the Selected items list box, then these will be moved in the specified direction, one step at a time until it reaches its limit. To reset the order of the columns in the order in which they appear in the dataset, click on the reset icon next to the Selected items list box. This will reset the columns in the view in the way the columns appear in the view.

To highlight items, Left-Click on the required item. To highlight mul-

tiple items in any of the list boxes, Left-Click and Shift-Left-Click will highlight all contiguous items, and Left-Click and Ctrl-Left-Click will add that item to the highlight elements.

The lower portion of the Columns panel provides a utility to highlight items in the Column Selector. You can either match by Name or by Experimental Factor (if specified). To match by Name, select Match By Name from the drop down list, enter a string in the Name text box and hit Enter. This will do a substring match with the Available List and the Selected list and highlight the matches. To match by Experiment Grouping, the Experiment Grouping information must be provided in the dataset. If this is available, the Experiment Grouping drop down will show the factors. The groups in each factor will be show in the Groups list box. Selecting specific Groups from the text box will highlight the corresponding items in the Available items and Selected items box above. These can be moved as explained above. By default, the match By Name is used.

Description: The title for the view and description or annotation for the view can be configured and modified from the description tab on the properties dialog. Right-Click on the view and open the Properties dialog. Click on the Description tab. This will show the Description dialog with the current Title and Description. The title entered here appears on the title bar of the particular view and the description if any will appear in the Legend window situated in the bottom of panel on the right. These can be changed changing the text in the corresponding text boxes and clicking OK. By default, if the view is derived from running an algorithm, the description will contain the algorithm and the parameters used.

3.9 The Matrix Plot View

The Matrix Plot is launched by Left-Click on Matrix Plot  icon on the main toolbar or from the View menu on the main menu bar. The Matrix Plot shows a matrix of pairwise 2D scatter plots for selected columns. The X-Axis and Y-Axis of each scatter plot are shown in the corresponding row and column. If columns are selected then the Matrix Plot is launched with the selected columns. If no column is selected, the Matrix plot is launched with the first three continuous columns in the dataset and is presented at a 3 x 3 scatter. If a Classlabel column is marked in the dataset, each Classlabel is

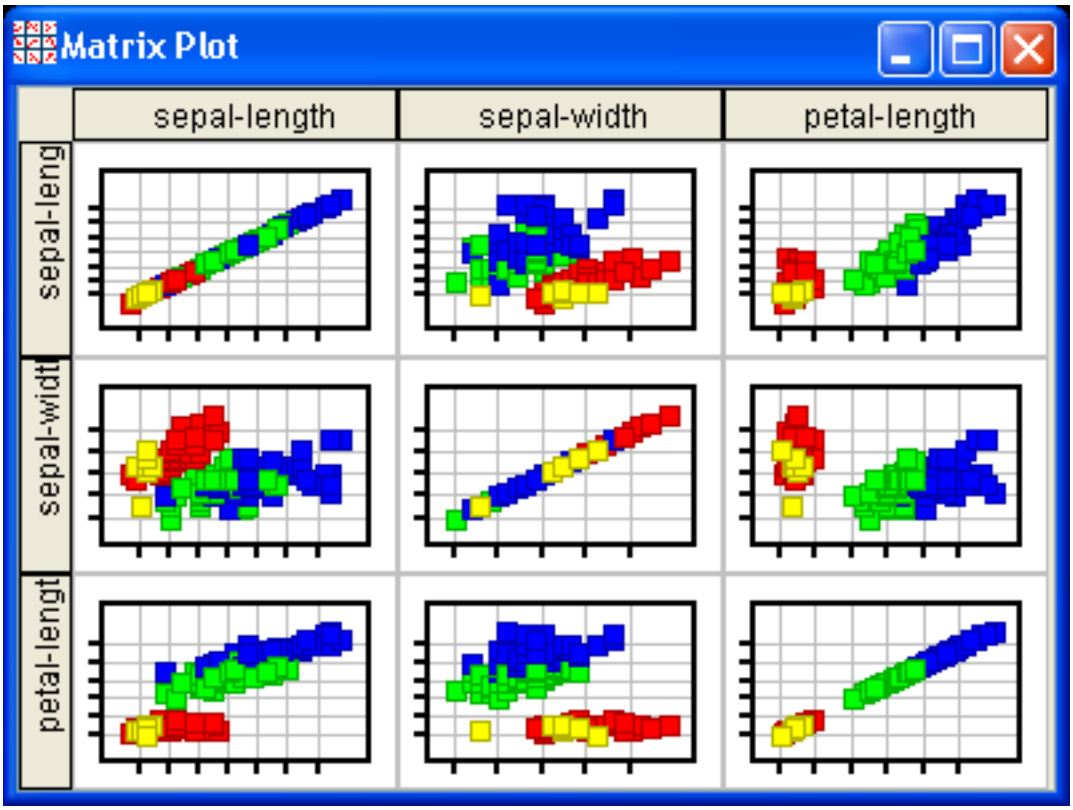


Figure 3.25: Matrix Plot

colored distinctly in the plot. If no class label column is marked, the Matrix plot is colored by the categorical column with the least number of categories in the active dataset. These colors can be changed from the **Properties** Dialog.

The main purpose of the Matrix Plot is to get an overview of the correlation between columns in the dataset, and detect columns that separate the data into different classes, if a Classlabel column is marked in the dataset.

A maximum of 10 columns can be shown in the Matrix Plot. If more than 10 columns are selected, only ten columns are projected into the Matrix Plot and other columns are ignored with a warning message. Moving the cursor onto the each plot displays the corresponding regression coefficient of the two axes in the ticker area of the tool. The Matrix plot is non-interactive and cannot be lassoed.

3.9.1 Matrix Plot Operations

The Matrix Plot operations are accessed from the main menu bar when the plot is the active windows. These operations are also available by Right-Click on the canvas of the Matrix Plot. Operations that are common to all views are detailed in the section Common Operations on Plot Views. Matrix Plot specific operations and properties are discussed below.

Selection Mode: The Matrix Plot supports only the Selection mode. Left-Click and dragging the mouse over the Matrix Plot draws a selection box and all points that intersect the selection box are selected and lassoed. To select additional elements, Ctrl-Left-Click and drag the mouse over the desired region. Ctrl-Left-Click toggles selection. This selected points will be unselected and unselected points will be added to the selection and lassoed.

3.9.2 Matrix Plot Properties

The Matrix Plot can be customized and configured from the properties dialog accessible from the Right-Click menu on the canvas of the Matrix plot, or from the view Properties icon on the main tool bar, or from the view menu on the main tool bar. The important properties of the scatter plot are all available for the Matrix plot. These are available in the Axis tab, the Visualization tab, the Rendering tab, the Columns tab and the description tab of the properties dialog and are detailed below.

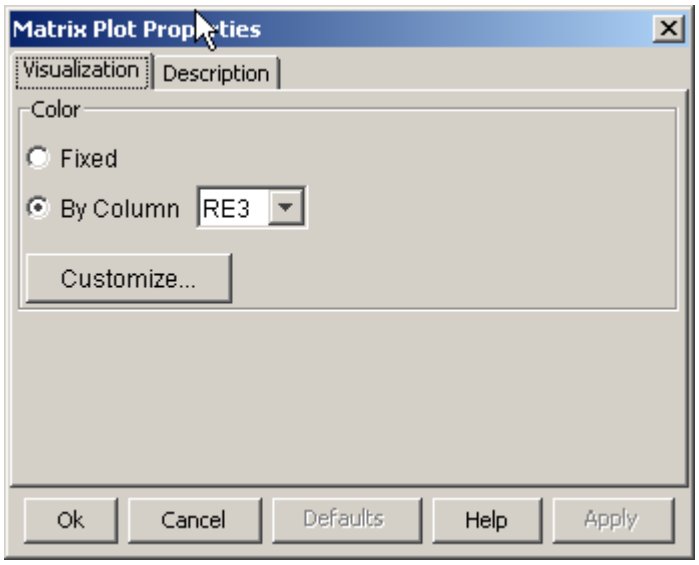


Figure 3.26: Matrix Plot Properties

Axis: The Axes on the Matrix Plot can be toggled to show or hide the grids, or show and hide the axis labels.

Visualization: The scatter plots can be configured to Color By any column of the active dataset, Shape By any categorical column of the dataset, and Size by any column of the dataset.

Rendering: The fonts on the Matrix Plot, the colors that occur on the Matrix Plot, the Offsets, the Page size of the view and the quality of the Matrix Plot can be be altered from the Rendering tab of the Properties dialog.

Fonts: All fonts on the plot, can be formatted and configured. To change the font in the view, Right-Click on the view and open the Properties dialog. Click on the Rendering tab of the Properties dialog. To change a Font, click on the appropriate drop-down box and choose the required font. To customize the font, click on the customize button. This will pop-up a dialog where you can set the font size and choose the font type as bold or italic.

Special Colors: All the colors that occur in the plot can be modified and configured. The plot Background Color, the Axis Color, the

Grid Color, the Selection Color, as well as plot specific colors can be set. To change the default colors in the view, Right-Click on the view and open the Properties dialog. Click on the Rendering tab of the Properties dialog. To change a color, click on the appropriate color bar. This will pop-up a Color Chooser. Select the desired color and click OK. This will change the corresponding color in the View.

Offsets: The left offset, right offset and the top offset and bottom offset of the plot can be modified and configured. These offsets may be need to be changed if the axis labels or axis titles are not completely visible in the plot, or if only the graph portion of the plot is required. To change the offsets, Right-Click on the view and open the Properties dialog. Click on the Rendering tab. To change plot offsets, move the corresponding slider, or enter an appropriate value in the text box provided. This will change the particular offset in the plot.

Page: The visualization page of the Matrix Plot can be configured to view a specific number scatter plots in the Matrix Plot. If there are more scatter plots in the Matrix plot than in the page, scroll bars appear and you can scroll to the other plot of the Matrix Plot.

Plot Quality: The quality of the plot can be enhanced to be anti-aliased. This will produce better points and will produce better prints of the Matrix Plot.

Columns: The Columns for the Matrix Plot can be chosen from the Columns tab of the Properties dialog.

The columns for visualization and the order in which the columns are visualized can be chosen and configured for the column selector. Right-Click on the view and open the properties dialog. Click on the columns tab. This will open the column selector panel. The column selector panel shows the Available items on the left-side list box and the Selected items on the right-hand list box. The items in the right-hand list box are the columns that are displayed in the view in the exact order in which they appear.

To move a columns from the Available list box to the Selected list box, highlight the required items in the Available items list box and click on the right arrow in between the list boxes. This will move the highlighted columns from the Available items list box to the bottom of

the Selected items list box. To move columns from the Selected items to the Available items, highlight the required items on the Selected items list box and click on the left arrow. This will move the highlight columns from the Selected items list box to the Available items list box in the exact position or order in which the column appears in the dataset.

You can also change the column ordering on the view by highlighting items in the Selected items list box and clicking on the up or down arrows. If multiple items are highlighted, the first click will consolidate the highlighted items (bring all the highlighted items together) with the first item in the specified direction. Subsequent clicks on the up or down arrow will move the highlighted items as a block in the specified direction, one step at a time until it reaches its limit. If only one item or contiguous items are highlighted in the Selected items list box, then these will be moved in the specified direction, one step at a time until it reaches its limit. To reset the order of the columns in the order in which they appear in the dataset, click on the reset icon next to the Selected items list box. This will reset the columns in the view in the way the columns appear in the view.

To highlight items, Left-Click on the required item. To highlight multiple items in any of the list boxes, Left-Click and Shift-Left-Click will highlight all contiguous items, and Left-Click and Ctrl-Left-Click will add that item to the highlight elements.

The lower portion of the Columns panel provides a utility to highlight items in the Column Selector. You can either match by Name or by Experimental Factor (if specified). To match by Name, select Match By Name from the drop down list, enter a string in the Name text box and hit Enter. This will do a substring match with the Available List and the Selected list and highlight the matches. To match by Experiment Grouping, the Experiment Grouping information must be provided in the dataset. If this is available, the Experiment Grouping drop down will show the factors. The groups in each factor will be show in the Groups list box. Selecting specific Groups from the text box will highlight the corresponding items in the Available items and Selected items box above. These can be moved as explained above. By default, the match By Name is used.

Description: The title for the view and description or annotation for the view can be configured and modified from the description tab on the

properties dialog. Right-Click on the view and open the Properties dialog. Click on the Description tab. This will show the Description dialog with the current Title and Description. The title entered here appears on the title bar of the particular view and the description if any will appear in the Legend window situated in the bottom of panel on the right. These can be changed changing the text in the corresponding text boxes and clicking OK. By default, if the view is derived from running an algorithm, the description will contain the algorithm and the parameters used.

3.10 Summary Statistics View

The Summary Statistics View is launched by Left-Click on Summary Statistics  icon on the main toolbar or from Menu bar on the main menu bar. Select columns in the Column Selection Dialog shown below. The Summary Statistics View can only be launched with continuous columns. If there are column selected in the dataset, the summary statistics view will be launched with the continuous columns in the selection. If there are no columns selected, the summary statistics view will be launched with all columns in the active dataset. This is Summary Statistics View is a table view and thus all operations and that are possible on a table are possible here. The Bar Chart can be customized and configured from the **Properties** dialog accessed from the Right-Click menu on the canvas of the Chart or from the icon on the tool bar.

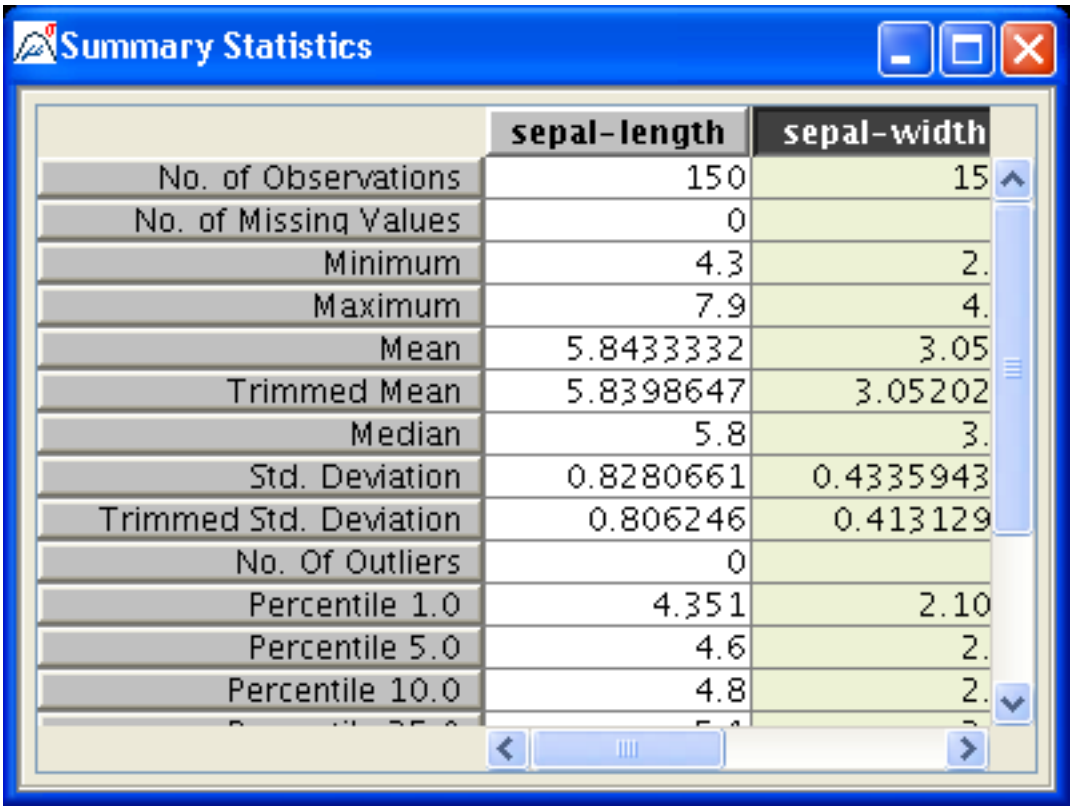
This view presents descriptive statistics information on every chosen column, and is useful to compare the distributions of different columns.

Note that the Summary statistics view will show only the continuous columns of the active dataset.

3.10.1 Summary Statistics Operations

The Operations on the Summary Statistics View is accessible from the menu on Right-Click on the canvas of the Summary Statistics View. Operations that are common to all views are detailed in the section Common Operations on Table Views above. In addition, some of the Summary Statistics View specific operations and the bar chart properties are explained below:

Column Selection: The Summary statistics View can be used to select columns, or any contiguous part of the dataset. The selected columns are lassoed in all the appropriate views.



	sepal-length	sepal-width
No. of Observations	150	15
No. of Missing Values	0	
Minimum	4.3	2.
Maximum	7.9	4.
Mean	5.8433332	3.05
Trimmed Mean	5.8398647	3.05202
Median	5.8	3.
Std. Deviation	0.8280661	0.4335943
Trimmed Std. Deviation	0.806246	0.413129
No. Of Outliers	0	
Percentile 1.0	4.351	2.10
Percentile 5.0	4.6	2.
Percentile 10.0	4.8	2.

Figure 3.27: Summary Statistics View

Columns can be selected by Left-Click in the column of interest. Ctrl-Left-Click selects subsequent columns and Shift-Left-Click consecutive set of columns. The current column selection on the bar chart usually determines the default set of selected columns used when launching any new view, executing commands or running algorithm. The selected columns will be lassoed in all relevant views and will be show selected in the lasso view.

Trellis: The Summary Statistics View can be trellised based on a trellis column. To trellis the Summary statistics View, click on Trellis on the Right-Click menu or click Trellis from the View menu. This will launch multiple Summary Statistics View in the same view based on the trellis column. By default the trellis will be launched with the categorical column with the least number of categories in the current dataset. You can change the trellis column by the properties of the trellis view.

Export As Text: The *Export* → *Text* option saves the tabular output to a tab-delimited file that can be opened in **ArrayAssist**.

3.10.2 Summary Statistics Properties

The Summary Statistics View Properties Dialog is accessible from Properties  icon on the main toolbar or by Right-Click on the Summary Statistics View and choosing **Properties** from the menu. The Summary Statistics View can be customized and configured from the Summary Statistics View properties.

Rendering: The rendering tab of the Summary Statistics View dialog allows you to configure and customize the fonts and colors that appear in the Summary Statistics View view.

Special Colors: All the colors in the Table can be modified and configured. You can change the Selection color, the Double Selection color, Missing Value cell color and the Background color in the table view. To change the default colors in the view, Right-Click on the view and open the Properties dialog. Click on the Rendering tab of the properties dialog. To change a color, click on the appropriate color bar. This will pop-up a Color Chooser. Select the desired color and click OK. This will change the corresponding color in the Table.

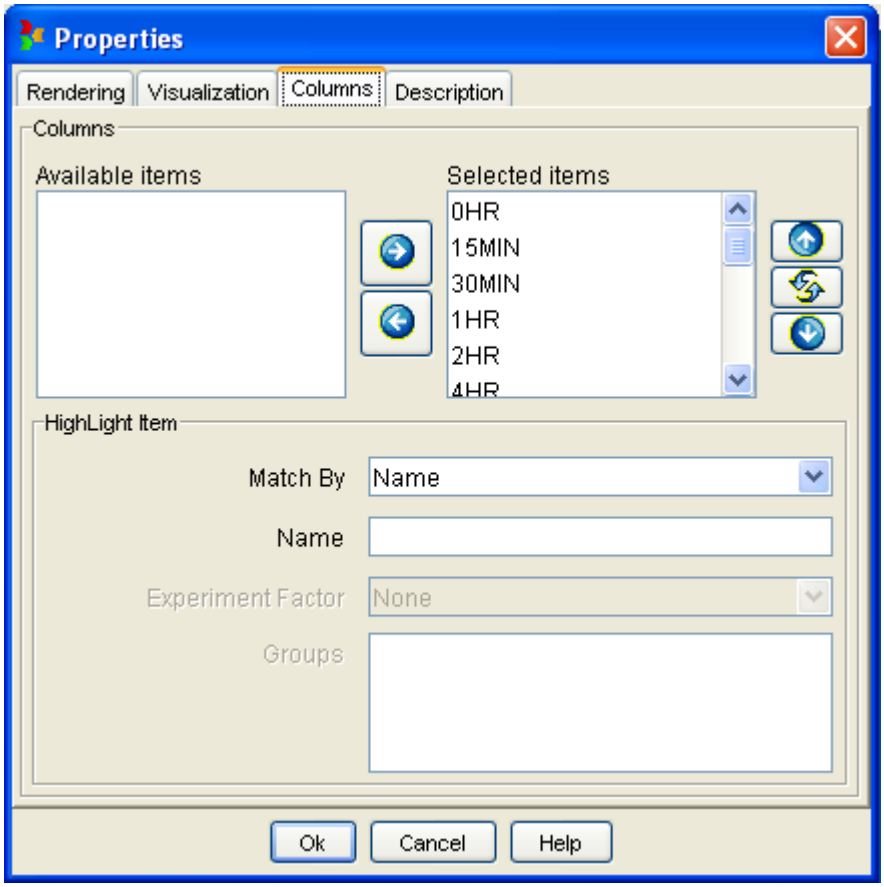


Figure 3.28: Summary Statistics Properties

Fonts: Fonts can be that occur in the table can be formatted and configured. You can set the fonts for Cell text, row Header and Column Header. To change the font in the view, Right-Click on the view and open the Properties dialog. Click on the Rendering tab of the Properties dialog. To change a Font, click on the appropriate drop-down box and choose the required font. To customise the font, click on the customise button. This will pop-up a dialog where you can set the font size and choose the font type as bold or italic.

Visualization: The display precision of decimal values in columns, the row height and the missing value text, and the facility to enable and disable sort are configured and customized by options in this tab.

Visualization: The visualization of the display precision of the numeric data in the table, the table cell size and the text for missing value can be configured. To change these, Right-Click on the table view and open the Properties dialog. Click on the visualization tab. This will open the Visualization panel.

To change the numeric precision. Click on the drop-down box and choose the desired precision. For decimal data columns you can choose between full precision and one to for decimal places, or representation in scientific notation. By default, full precision is displayed.

You can set the row height of the table, by entering a integer value in the text box and pressing Enter. This will change the row height in the table. By default the row height is set to 16.

You can enter any a text to show missing values. All missing values in the table will be represented by the entered value and missing values can be easily identified. By default all the missing value text is set to an empty string.

You can also enable and disable sorting on any column of the table by checking or unchecking the check box provided. By default, sort is enabled in the table. To sort the table on any column, click on the column header. This will sort the all rows of the table based on the values in the sort column. This will also mark the sorted column with an icon to denote the sorted column. The first click on the column header will sort the column in the ascending order, the second click on the column header will sort the column in the descending order, and clicking the sorted column the third time would reset the sort.

Columns: The order of the columns in the Summary Statistics View can be changed by changing the order in the Columns tab in the Properties Dialog.

The columns for visualization and the order in which the columns are visualized can be chosen and configured for the column selector. Right-Click on the view and open the properties dialog. Click on the columns tab. This will open the column selector panel. The column selector panel shows the Available items on the left-side list box and the Selected items on the right-hand list box. The items in the right-hand list box are the columns that are displayed in the view in the exact order in which they appear.

To move a columns from the Available list box to the Selected list box, highlight the required items in the Available items list box and click on the right arrow in between the list boxes. This will move the highlighted columns from the Available items list box to the bottom of the Selected items list box. To move columns from the Selected items to the Available items, highlight the required items on the Selected items list box and click on the left arrow. This will move the highlight columns from the Selected items list box to the Available items list box in the exact position or order in which the column appears in the dataset.

You can also change the column ordering on the view by highlighting items in the Selected items list box and clicking on the up or down arrows. If multiple items are highlighted, the first click will consolidate the highlighted items (bring all the highlighted items together) with the first item in the specified direction. Subsequent clicks on the up or down arrow will move the highlighted items as a block in the specified direction, one step at a time until it reaches its limit. If only one item or contiguous items are highlighted in the Selected items list box, then these will be moved in the specified direction, one step at a time until it reaches its limit. To reset the order of the columns in the order in which they appear in the dataset, click on the reset icon next to the Selected items list box. This will reset the columns in the view in the way the columns appear in the view.

To highlight items, Left-Click on the required item. To highlight multiple items in any of the list boxes, Left-Click and Shift-Left-Click will highlight all contiguous items, and Left-Click and Ctrl-Left-Click will add that item to the highlight elements.

The lower portion of the Columns panel provides a utility to highlight items in the Column Selector. You can either match by Name or by Experimental Factor (if specified). To match by Name, select Match By Name from the drop down list, enter a string in the Name text box and hit Enter. This will do a substring match with the Available List and the Selected list and highlight the matches. To match by Experiment Grouping, the Experiment Grouping information must be provided in the dataset. If this is available, the Experiment Grouping drop down will show the factors. The groups in each factor will be show in the Groups list box. Selecting specific Groups from the text box will highlight the corresponding items in the Available items and Selected items box above. These can be moved as explained above. By default, the match By Name is used.

Description: The title for the view and description or annotation for the view can be configured and modified from the description tab on the properties dialog. Right-Click on the view and open the Properties dialog. Click on the Description tab. This will show the Description dialog with the current Title and Description. The title entered here appears on the title bar of the particular view and the description if any will appear in the Legend window situated in the bottom of panel on the right. These can be changed changing the text in the corresponding text boxes and clicking OK. By default, if the view is derived from running an algorithm, the description will contain the algorithm and the parameters used.

3.11 The Box Whisker Plot

The Box Whisker Plot is launched by Left-Click on Box Whisker Plot  icon on the tool bar or from View menu on the main menu bar. The Box Whisker Plot presents the distribution of the values in any column of the dataset. Each column is represented by two figures, the box whisker of the points in the column and the a density scatter of points in the column next to it. The box whisker shows the median in the middle of the box, the 25th quartile and the 75th quartile. The whiskers are extensions of the box, snapped to the point within 1.5 times the interquartile. The points outside the whiskers are plotted as they are, but in a different color and could normally be considered the outliers. The density plot next to the box whisker is a plot of all points in the column. This will give visual representation of the distribution and the density of the values in the column.

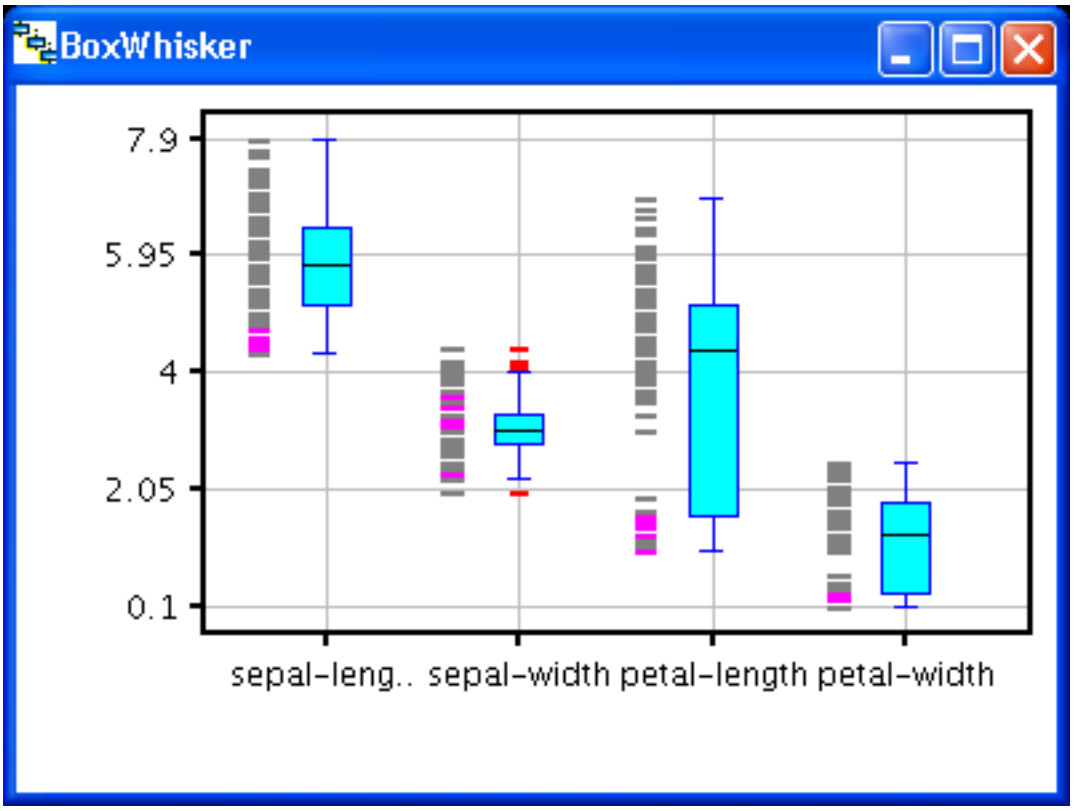


Figure 3.29: Box Whisker Plot

The operations on the box whisker plot are similar to operations on all plots and will be discussed below. The box whisker plot can be customized and configured from the **Properties** dialog. If a columns are selected in the spreadsheet, the box whisker plot is be launched with the continuous columns in the selection. If no columns are selected, then the box whisker will be launched with all continuous columns in the active dataset.

3.11.1 Box Whisker Operations

The Box Whisker operations are accessed from the toolbar menu when the plot is the active window. These operations are also available by Right-Click on the canvas of the Box Whisker. Operations that are common to all views are detailed in the section Common Operations on Plot Views. Box Whisker specific operations and properties are discussed below.

Selection Mode: The Selection on the Box Whisker plot is confined to only one column of plot. This is so because the box whisker plot contains box whiskers for many columns and each of them contain all the rows in the active dataset. Thus selection has to be confined to only to one column in the plot. The Box Whisker only supports the selection mode. Thus, Left-Click and dragging the mouse over the box whisker plot confines the selection box to only one column. The points in this selection box are highlighted in the density plot of that particular column and are also lassoed highlighted in the density plot of all other columns. Left-Click and dragging, and Shift-Left-Click and dragging selects elements and Ctrl-Left-Click toggles selection like in any other plot and appends to the selected set of elements.

Trellis: The box whisker can be trellised based on a trellis column. To trellis the box whisker, click on Trellis on the Right-Click menu or click Trellis from the View menu. This will launch multiple box whisker in the same view based on the trellis column. By default the trellis will be launched with the categorical column with the least number of categories in the current dataset. You can change the trellis column by the properties of the trellis view.

3.11.2 Box Whisker Properties

The Box Whisker Plot offers a wide variety of customization and configuration of the plot from the Properties dialog. These customizations appear

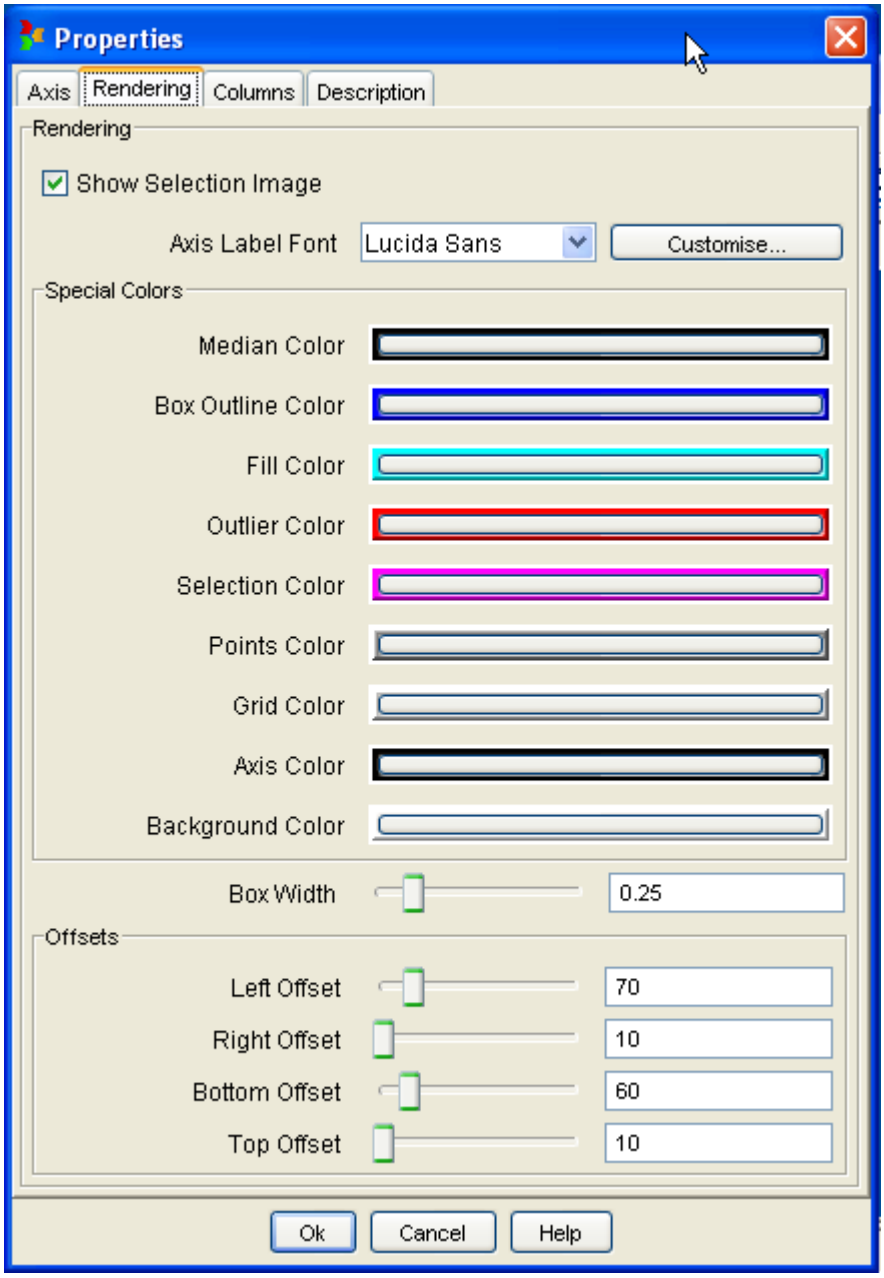


Figure 3.30: Box Whisker Properties

in three different tabs on the Properties window, labelled Axis, Rendering, Columns, and Description.

Axis: The grids, axes labels, and the axis ticks of the plots can be configured and modified. To modify these, Right-Click on the view, and open the Properties dialog. Click on the Axis tab. This will open the axis dialog. The plot can be drawn with or without the grid lines by clicking on the show grids option. The tics and axis labels are automatically computed for the plot and show on the plot. You can show or remove the axis labels by clicking on the Show Axis Labels check box. The number of ticks on the axis are automatically computed to a show equal intervals between the minimum and maximum and displayed. You can increase the number of ticks displayed on the plot by moving the Axis Ticks slider. For continuous data columns, you can double the number of ticks shown by moving the slider to the maximum. For categorical columns, if the number of categories are less than ten, all the categories are show and moving the slider does not increase the number of tics.

Rendering: The Box Whisker Plot allows all aspects of the view to be customizing the configured. The fonts, the colors, the offsets, etc can be configured.

Show Selection Image: The Show Selection Image, shows the density of points for each column of the box whisker plot. This is used for selection of points. For large datasets and for many columns this may take a lot of resources. You can choose to remove the density plot next to each box whisker by unckecking the check box provided.

Fonts: All fonts on the plot, can be formatted and configured. To change the font in the view, Right-Click on the view and open the Properties dialog. Click on the Rendering tab of the Properties dialog. To change a Font, click on the appropriate drop-down box and choose the required font. To customize the font, click on the customize button. This will pop-up a dialog where you can set the font size and choose the font type as bold or italic.

Special Colors: All the colors on the box whisker can be configured and customized.

All the colors that occur in the plot can be modified and configured. The plot Background Color, the Axis Color, the Grid

Color, the Selection Color, as well as plot specific colors can be set. To change the default colors in the view, Right-Click on the view and open the Properties dialog. Click on the Rendering tab of the Properties dialog. To change a color, click on the appropriate color bar. This will pop-up a Color Chooser. Select the desired color and click OK. This will change the corresponding color in the View.

Box Width: The box width of the box whisker plots can be changed by moving the slider provided. The default is set to 0.25 of the width provided to each column of the box whisker plot.

Offsets: The left offset, right offset and the top offset and bottom offset of the plot can be modified and configured. These offsets may be need to be changed if the axis labels or axis titles are not completely visible in the plot, or if only the graph portion of the plot is required. To change the offsets, Right-Click on the view and open the Properties dialog. Click on the Rendering tab. To change plot offsets, move the corresponding slider, or enter an appropriate value in the text box provided. This will change the particular offset in the plot.

Columns: The columns drawn in the Box Whisker Plot and the order of columns in the Box whisker Plot can be changed from the Columns tab in the Properties Dialog.

The columns for visualization and the order in which the columns are visualized can be chosen and configured for the column selector. Right-Click on the view and open the properties dialog. Click on the columns tab. This will open the column selector panel. The column selector panel shows the Available items on the left-side list box and the Selected items on the right-hand list box. The items in the right-hand list box are the columns that are displayed in the view in the exact order in which they appear.

To move a columns from the Available list box to the Selected list box, highlight the required items in the Available items list box and click on the right arrow in between the list boxes. This will move the highlighted columns from the Available items list box to the bottom of the Selected items list box. To move columns from the Selected items to the Available items, highlight the required items on the Selected items list box and click on the left arrow. This will move the highlight columns from the Selected items list box to the Available items list

box in the exact position or order in which the column appears in the dataset.

You can also change the column ordering on the view by highlighting items in the Selected items list box and clicking on the up or down arrows. If multiple items are highlighted, the first click will consolidate the highlighted items (bring all the highlighted items together) with the first item in the specified direction. Subsequent clicks on the up or down arrow will move the highlighted items as a block in the specified direction, one step at a time until it reaches its limit. If only one item or contiguous items are highlighted in the Selected items list box, then these will be moved in the specified direction, one step at a time until it reaches its limit. To reset the order of the columns in the order in which they appear in the dataset, click on the reset icon next to the Selected items list box. This will reset the columns in the view in the way the columns appear in the view.

To highlight items, Left-Click on the required item. To highlight multiple items in any of the list boxes, Left-Click and Shift-Left-Click will highlight all contiguous items, and Left-Click and Ctrl-Left-Click will add that item to the highlight elements.

The lower portion of the Columns panel provides a utility to highlight items in the Column Selector. You can either match by Name or by Experimental Factor (if specified). To match by Name, select Match By Name from the drop down list, enter a string in the Name text box and hit Enter. This will do a substring match with the Available List and the Selected list and highlight the matches. To match by Experiment Grouping, the Experiment Grouping information must be provided in the dataset. If this is available, the Experiment Grouping drop down will show the factors. The groups in each factor will be show in the Groups list box. Selecting specific Groups from the text box will highlight the corresponding items in the Available items and Selected items box above. These can be moved as explained above. By default, the match By Name is used.

Description: The title for the view and description or annotation for the view can be configured and modified from the description tab on the properties dialog. Right-Click on the view and open the Properties dialog. Click on the Description tab. This will show the Description dialog with the current Title and Description. The title entered here appears on the title bar of the particular view and the description

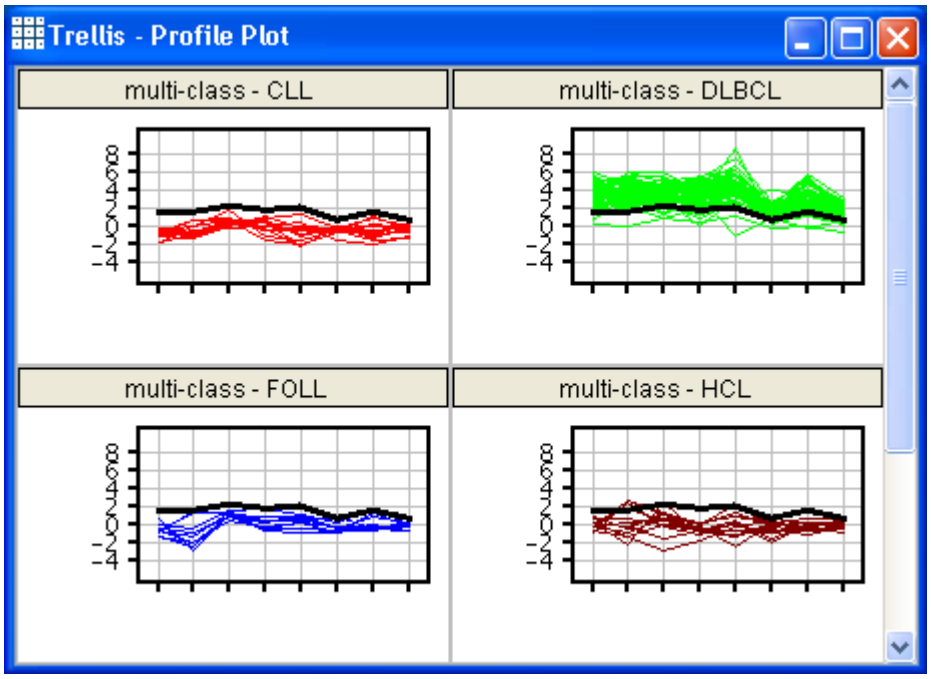


Figure 3.31: Trellis of Profile Plot

if any will appear in the Legend window situated in the bottom of panel on the right. These can be changed changing the text in the corresponding text boxes and clicking OK. By default, if the view is derived from running an algorithm, the description will contain the algorithm and the parameters used.

3.12 Trellis

The Trellis View is a derived view. The Trellis view can be derived and launched from the Spreadsheet, the Scatter Plot, the Profile Plot, the Histogram, the Summary Statistics, and the Bar Chart view. To launch the Trellis view on any of the above views, Right-Click on the canvas of the view and select Trellis, or choose Trellis from the Views menu on the main menu bar with the active view being one of the above.

The Trellis view will split the view on which Trellis is launched, into multiple views based on a categorical column. This is done by dividing the dataset into different groups based upon the categories in the trellis by

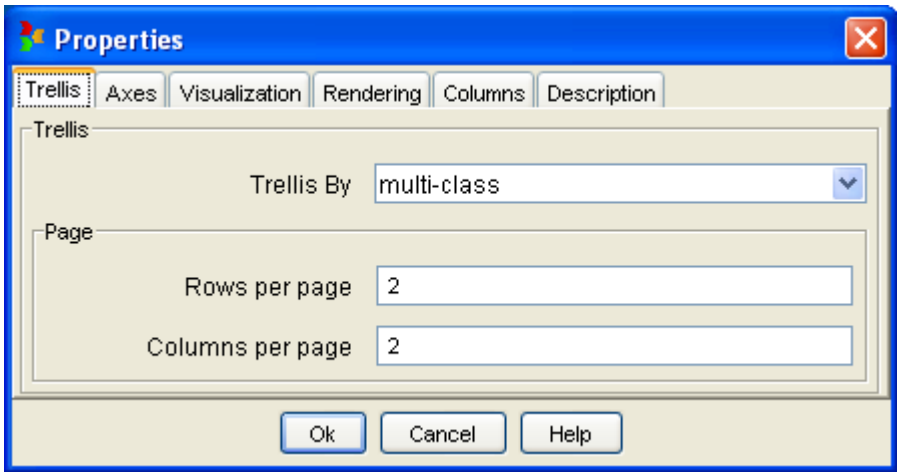


Figure 3.32: Trellis Properties

column and launch multiple views, one for each category in the trellis by column. By default, trellis will be launched with the trellis by column as the categorical column with the least number of categories. Trellis can be launched with a maximum of 50 categories in the trellis by column. If the dataset does not have a categorical column with less than 50 categories, an error dialog is displayed. The Trellis column can be changed from the Properties dialog of the Trellis view.

3.12.1 Trellis View Operations

The operations on the Trellis View are accessed from the toolbar menu when the plot is the active window. These operations are also available by Right-Click on the canvas of the Trellis View. Operations that are common to all views are detailed in the section Common Operations on Plot Views. The Trellis View supports all the operations of the view from which the Trellis is launched. Thus if the Spreadsheet is trellised, then all operations on the Spreadsheet are supported by the Trellis View.

3.12.2 Trellis Poperties

The Trellis Properties are accessed from Right-Click on the canvas of the Trellis View. The Properties on the Trellis View are derived from the properties of the parent view. Thus most of the Properties of the parent view are

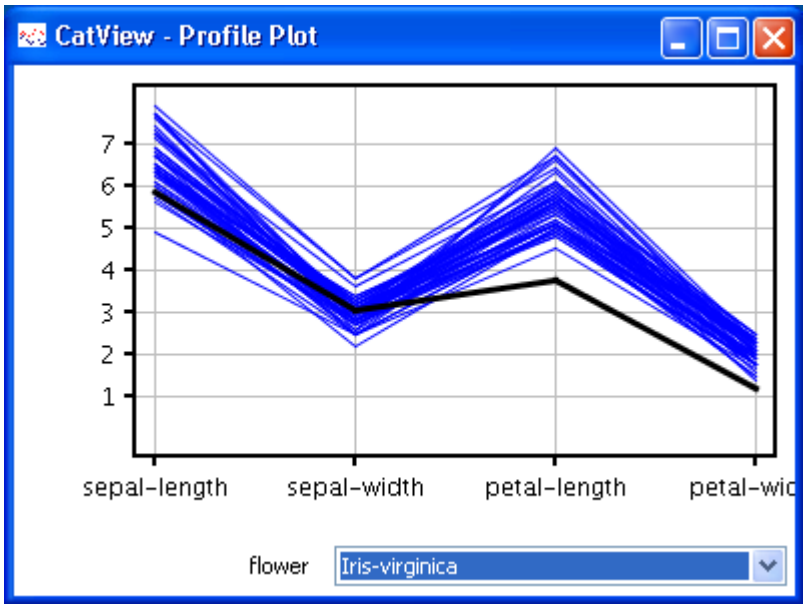


Figure 3.33: CatView of Scatter Plot

available on the Trellis View and the unavailable properties will be disabled. In addition the following options are available on the Trellis View to configure and customize the Trellis View under the Trellis tab of the Properties dialog.

Trellis By: The trellis By columns for the Trellis view can be changed to any categorical column of the active dataset displayed the drop down list. By default, the Trellis column is the column with the least number of categories. Note that the Trellis can be launched with a maximum of 50 categories.

Page Size: The visualization page of the trellis Plot can be configured to view a specific number of views. The number of rows and number of columns in each page of the view can be set. If there are more Trellis views than can be shown in one page, scroll bars appear on the trellis view that can be scrolled to view multiple pages.

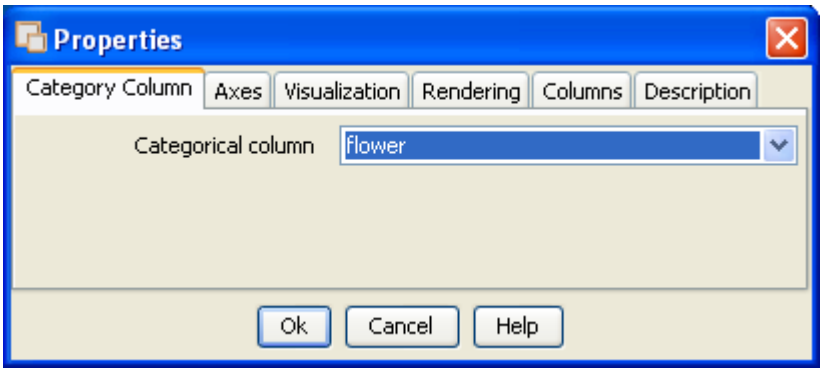


Figure 3.34: CatView Properties

3.13 CatView

The CatView is a derived view. The CatView can be derived and launched from Spreadsheet, the Scatter Plot, the Profile Plot, the Histogram, the Summary Statistics, and the Bar Chart view. To launch the CatView on any of the above views, Right-Click on the canvas of the view and select CatView.

The CatView will launch a view of the parent view with one of the category values of the categorical column. The view only shows the data corresponding a single categorical value in the chosen column. By default, the CatView will be launched with the categorical column with the least number of categories. The category values in the column are shown in the drop-down of the view and can be changed.

3.13.1 CatView Operations

The operations on the CatView are accessed from the toolbar menu when the plot is the active window. These operations are also available by Right-Click on the canvas of the CatView. Operations that are common to all views are detailed in the section Common Operations on Plot Views. The CatView supports all the operations of the view from which the CatView is launched. Thus if a CatView is launched on the Scatter plot, then all operations on the Scatter plot are supported by the CatView.

3.13.2 CatView Poperties

The CatView Properties are accessed from Right-Click on the canvas of the CatView. The Properties on the CatView are derived from the properties of the parent view. Thus most of the Properties of the parent view are available on the CatView and the unavailable properties will be disabled. In addition the following options are available on the CatView to configure and customize the CatView under the Category Column tab of the Properties dialog.

Category Column: The category column for the CatView can chosen and changed from the drop-down list of categorical columns available in the current active dataset. By default, the categorical column with the least number of categories will be chosen as a categorical column for the view.

3.14 The Lasso View

The *Lasso* view shows actual data details of the rows selected in any linked view. A subset of columns to be displayed can be set from the view's Properties. Columns in this window can be stretched or shuffled and this configuration is maintained as various selections are performed, allowing the user to concentrate on values in a few columns.

3.14.1 Lasso Properties

The properties of the Lasso window is accessible by Right-Click on the Lasso Window. This allows customizing columns required to be shown in the Lasso Window. By default all the columns are shown in the Lasso Window.

Rendering: The rendering tab of the Lasso Window dialog allows you to configure and customize the fonts and colors that appear in the Lasso Window view.

Special Colors: All the colors in the Table can be modified and configured. You can change the Selection color, the Double Selection color, Missing Value cell color and the Background color in the table view. To change the default colors in the view, Right-Click on the view and open the Properties dialog. Click on the Rendering tab of the properties dialog. To change a color, click on the appropriate color bar. This will pop-up a Color Chooser. Select the

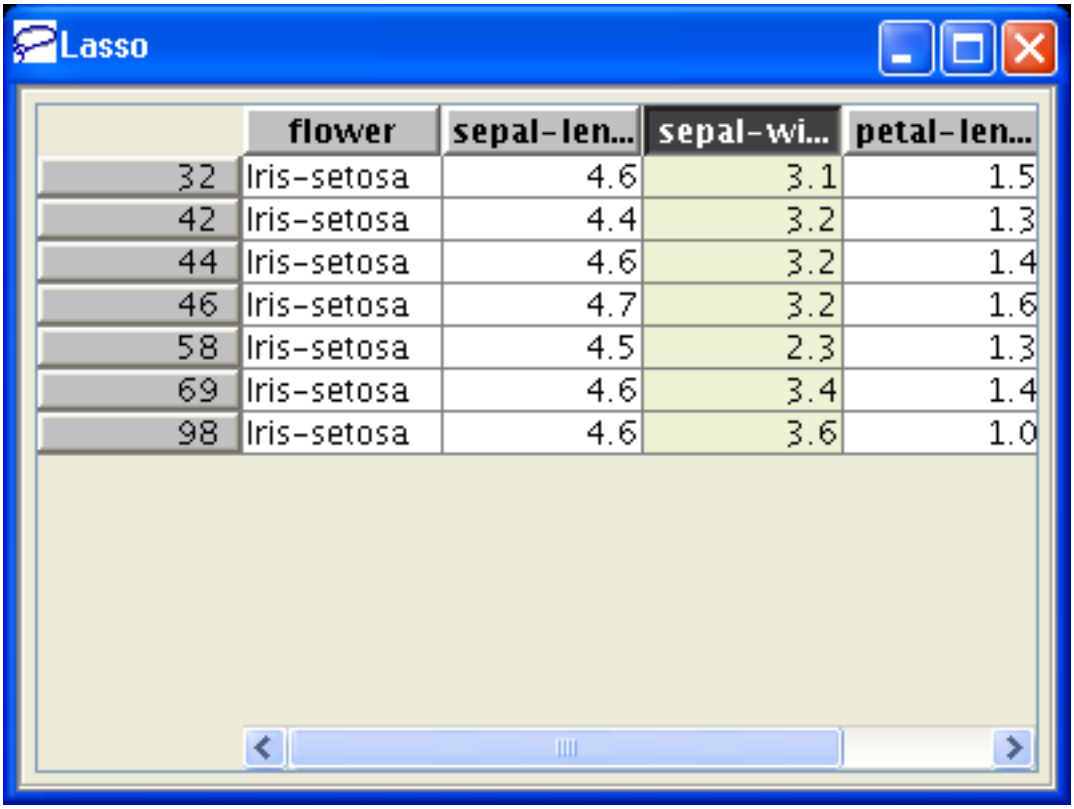


Figure 3.35: The Lasso Window

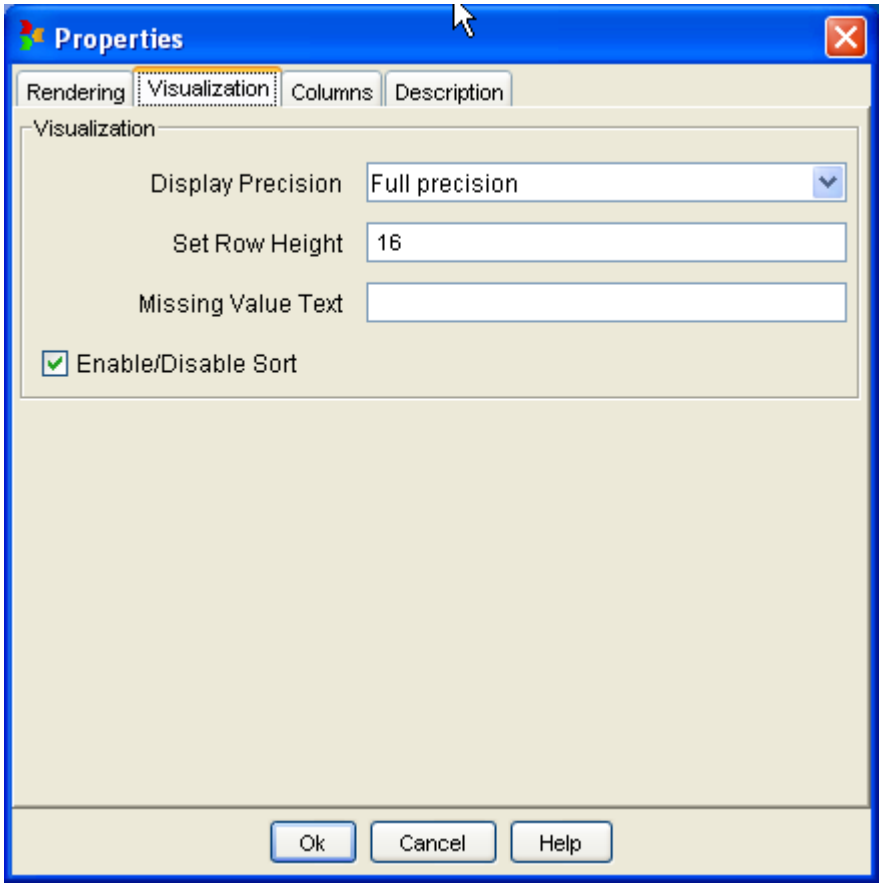


Figure 3.36: The Lasso Window Properties

desired color and click OK. This will change the corresponding color in the Table.

Fonts: Fonts can be that occur in the table can be formatted and configured. You can set the fonts for Cell text, row Header and Column Header. To change the font in the view, Right-Click on the view and open the Properties dialog. Click on the Rendering tab of the Properties dialog. To change a Font, click on the appropriate drop-down box and choose the required font. To customise the font, click on the customise button. This will pop-up a dialog where you can set the font size and choose the font type as bold or italic.

Visualization: The display precision of decimal values in columns, the row height and the missing value text, and the facility to enable and disable sort are configured and customized by options in this tab.

Visualization: The visualization of the display precision of the numeric data in the table, the table cell size and the text for missing value can be configured. To change these, Right-Click on the table view and open the Properties dialog. Click on the visualization tab. This will open the Visualization panel.

To change the numeric precision. Click on the drop-down box and choose the desired precision. For decimal data columns you can choose between full precision and one to for decimal places, or representation in scientific notation. By default, full precision is displayed.

You can set the row height of the table, by entering a integer value in the text box and pressing Enter. This will change the row height in the table. By default the row height is set to 16.

You can enter any a text to show missing values. All missing values in the table will be represented by the entered value and missing values can be easily identified. By default all the missing value text is set to an empty string.

You can also enable and disable sorting on any column of the table by checking or unchecking the check box provided. By default, sort is enabled in the table. To sort the table on any column, click on the column header. This will sort the all rows of the table based on the values in the sort column. This will also mark the sorted column with an icon to denote the sorted column. The first click on the column header will sort the column in the ascending order, the second click on

the column header will sort the column in the descending order, and clicking the sorted column the third time would reset the sort.

Columns: The order of the columns in the Lasso Window can be changed by changing the order in the Columns tab in the Properties Dialog.

The columns for visualization and the order in which the columns are visualized can be chosen and configured for the column selector. Right-Click on the view and open the properties dialog. Click on the columns tab. This will open the column selector panel. The column selector panel shows the Available items on the left-side list box and the Selected items on the right-hand list box. The items in the right-hand list box are the columns that are displayed in the view in the exact order in which they appear.

To move a columns from the Available list box to the Selected list box, highlight the required items in the Available items list box and click on the right arrow in between the list boxes. This will move the highlighted columns from the Available items list box to the bottom of the Selected items list box. To move columns from the Selected items to the Available items, highlight the required items on the Selected items list box and click on the left arrow. This will move the highlight columns from the Selected items list box to the Available items list box in the exact position or order in which the column appears in the dataset.

You can also change the column ordering on the view by highlighting items in the Selected items list box and clicking on the up or down arrows. If multiple items are highlighted, the first click will consolidate the highlighted items (bring all the highlighted items together) with the first item in the specified direction. Subsequent clicks on the up or down arrow will move the highlighted items as a block in the specified direction, one step at a time until it reaches its limit. If only one item or contiguous items are highlighted in the Selected items list box, then these will be moved in the specified direction, one step at a time until it reaches its limit. To reset the order of the columns in the order in which they appear in the dataset, click on the reset icon next to the Selected items list box. This will reset the columns in the view in the way the columns appear in the view.

To highlight items, Left-Click on the required item. To highlight multiple items in any of the list boxes, Left-Click and Shift-Left-Click will highlight all contiguous items, and Left-Click and Ctrl-Left-Click will

add that item to the highlight elements.

The lower portion of the Columns panel provides a utility to highlight items in the Column Selector. You can either match by Name or by Experimental Factor (if specified). To match by Name, select Match By Name from the drop down list, enter a string in the Name text box and hit Enter. This will do a substring match with the Available List and the Selected list and highlight the matches. To match by Experiment Grouping, the Experiment Grouping information must be provided in the dataset. If this is available, the Experiment Grouping drop down will show the factors. The groups in each factor will be show in the Groups list box. Selecting specific Groups from the text box will highlight the corresponding items in the Available items and Selected items box above. These can be moved as explained above. By default, the match By Name is used.

Description: The title for the view and description or annotation for the view can be configured and modified from the description tab on the properties dialog. Right-Click on the view and open the Properties dialog. Click on the Description tab. This will show the Description dialog with the current Title and Description. The title entered here appears on the title bar of the particular view and the description if any will appear in the Legend window situated in the bottom of panel on the right. These can be changed changing the text in the corresponding text boxes and clicking OK. By default, if the view is derived from running an algorithm, the description will contain the algorithm and the parameters used.

Chapter 4

Dataset Operations

4.1 Dataset Operations

All operations available on the dataset are listed below. These are organized into three categories: Column operations, Row operations and Dataset operations. Note that when column operations are performed you can choose to either append columns to the current dataset or you can choose to create a new child dataset with the transformed columns. Often, you may not like to clutter up the dataset with all transformed columns, rather you would like to focus on the transfromed dataset in your downstream analysis. In such situations you convenienetly create a child dataset. This is default output option in all the command operations.

4.1.1 Column Commands

The following column operations are available in the **Data** menu. All column operations allow column selection in the dialog. By default if no columns are selected in the active dataset, all columns will be selected and if some columns are selected in the active dataset, the column command will be launched with the selected columns.

The default option option is to create a child dataset. You can change the default name of the child dataset. Note that you cannot change the name of the child dataset after it has been created.

If you want to see all the columns in the dataset, the master dataset at the root of the navigator window will contain all the columns in the current project.

Logarithm: Use this to find logarithms of values in selected columns to bases 2, 10 or e; columns can be selected from the Select Columns panel

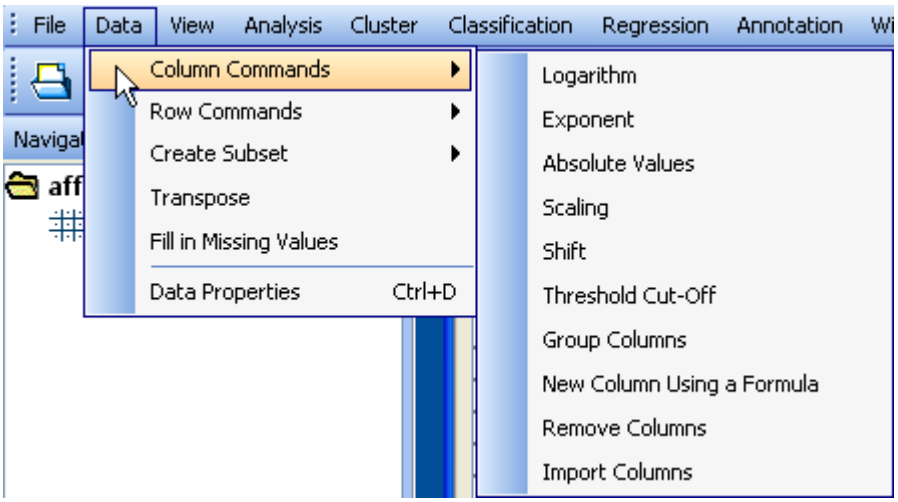


Figure 4.1: Data Menu

in the dialog box or using column selections on the spreadsheet. To select columns from the Select Columns panel, select the appropriate columns and then move them to the panel on the right. If numeric columns have been selected on the spreadsheet, these will appear on the panel on the right. Logarithms of selected columns are computed and appended to the dataset. Logarithms of non-negative or Missing Values will result in a Missing Value.

Exponent: Use this to exponentiate columns to bases 2, 10 or e. The usage is similar to **Logarithm**. Note that exponentiation could result in large values which when beyond a certain threshold will be treated as Missing Values.

Absolute: Use this to find absolute values of numerical data in selected columns. The usage is similar to **Logarithm**. This operation will compute the absolute value of all the values in the selected columns.

Scale: Use this to scale values in selected columns up or down by specified amounts. This multiplies or divides the values in the selected columns by the value entered in the dialog. The usage is similar to **Logarithm**.

Shift:

Use this function to shift all values by a constatnt positive value or a constatnt negative value in the selected columns by a constatnt. You can enter the constant float value in the text box. This will create a new column adding or subtracting the specified offset value to all values in the column. The usage and options are similar to **Logarithm**

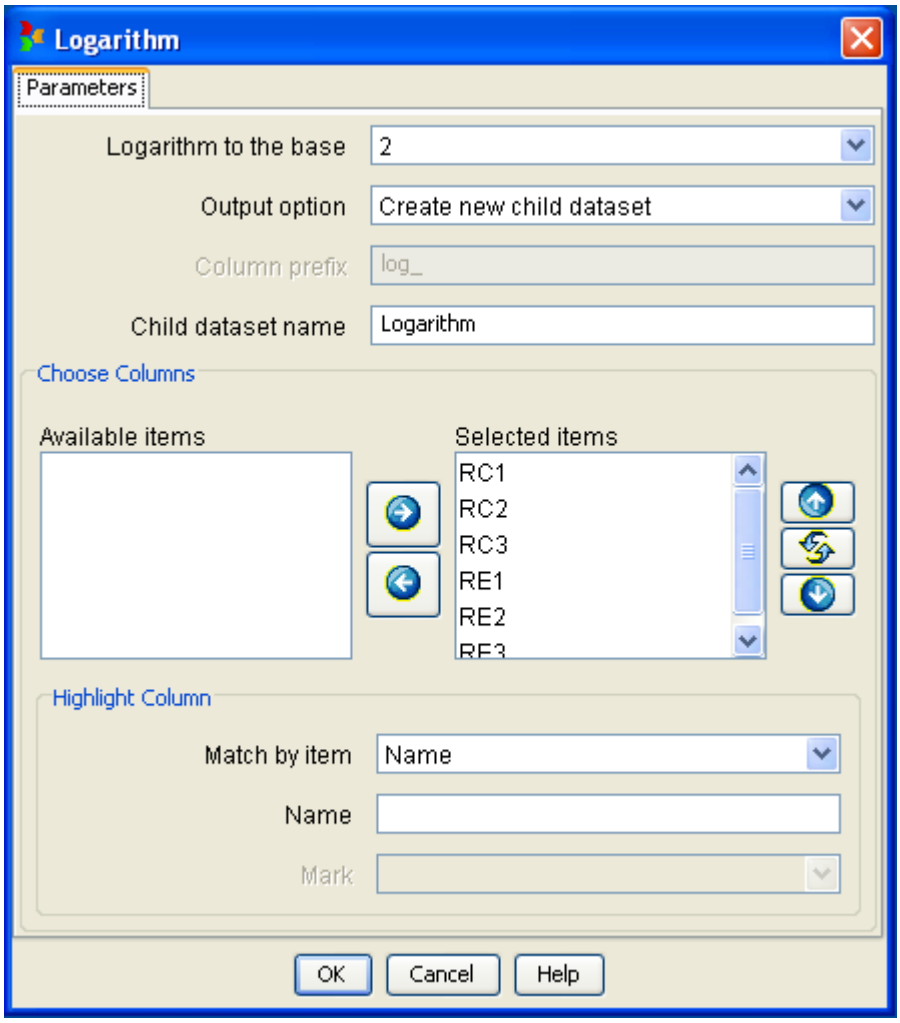


Figure 4.2: Logarithm Command

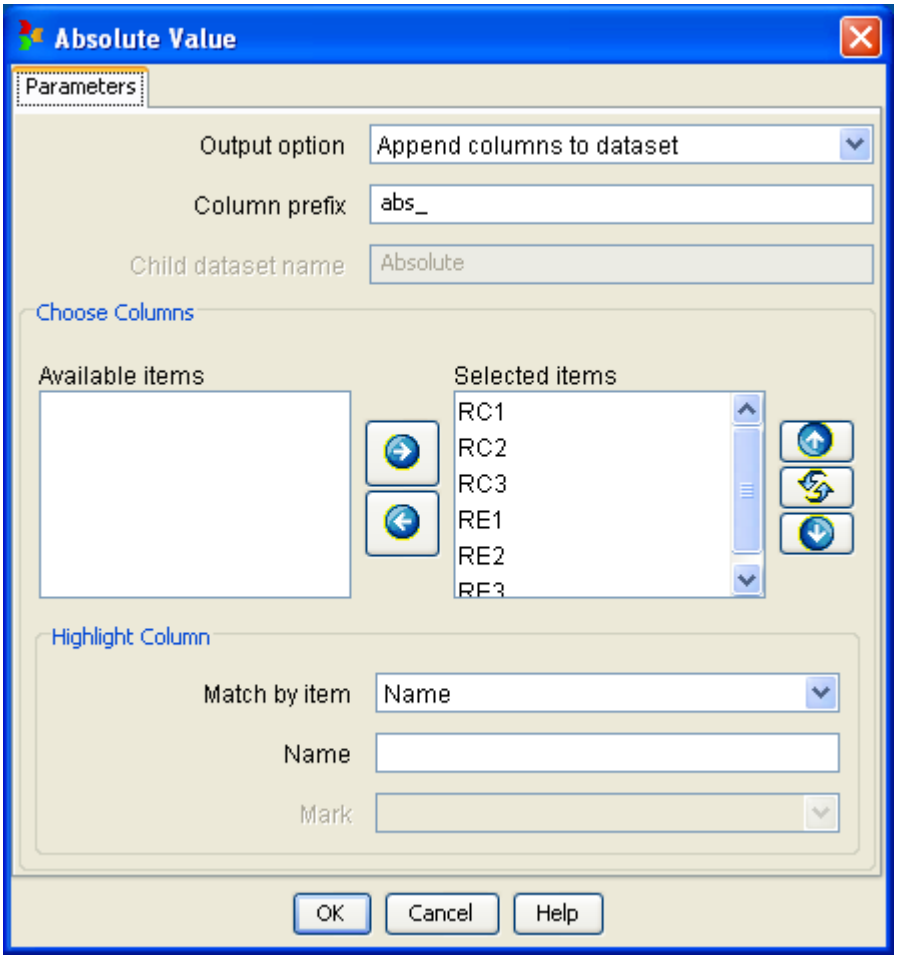


Figure 4.3: Absolute Command

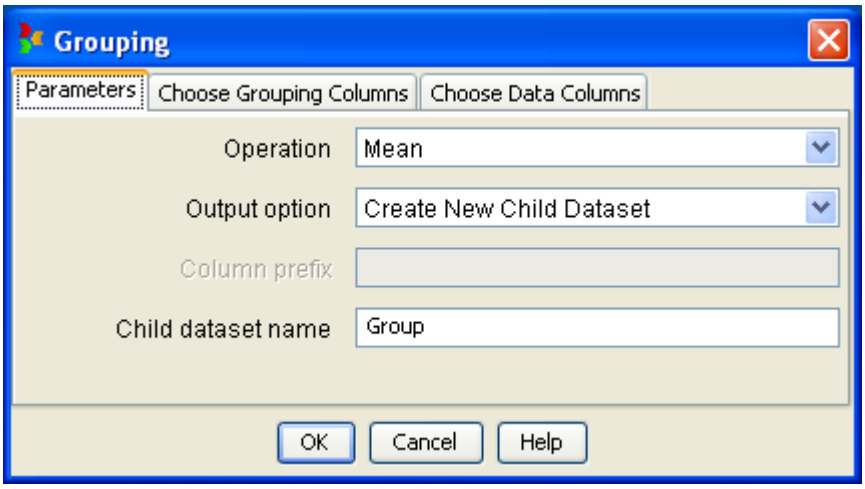


Figure 4.4: Append Column by Grouping

Threshold: Use this to threshold values in selected columns from above and/or below. The usage is similar to **Logarithm**. Values above the *max threshold* value, if specified are set to this value, as are values below the *min threshold*. This function is used to remove negative values from the data, in case logarithms need to be taken.

Group Columns: This facility is best explained with an example. Suppose you have a dataset where each row corresponds to a patient and each patient is given exactly one of three drugs, A, B or C; so the dataset has a column called *drug* only 3 distinct values, say A, B and C. Further, you have a column called *size* which stores a measurement for each patient.

Suppose you select *drug* as the grouping column and *size* as the data column in the interface shown above. Further, you choose *mean* as the grouping function. Then the new column that is added will contain for each patient given drug A, the average *size* over all patients given A, and likewise for patients given drugs B and C.

In general you could choose multiple grouping columns (in which case, groups will comprise rows which have identical values in ALL of these columns). You can also choose multiple data columns (in which case, a new column will be added for each data column chosen). Further, you can choose a function other than mean: the choices available are median, standard deviation, variance, standard error of mean (which is just standard deviation divided by the square root of the number of samples in a group),

range (the maximum-minimum value in the group), rank (the rank of each value among the values in a group), count, sum, maximum and minimum.

Finally, you can create new dataset with the columns grouped with a grouping column or you can append columns to the dataset with a specified a column prefix. When multiple data columns are chosen, multiple columns will be appended to the dataset and it would not be feasible for the user to provide a name for each such column. Instead a column prefix is sought; the new columns will have this prefix along with the original column names.

Create New Column using Formula:

A variety of mathematical, statistical and pattern matching functions are available here. These are grouped under different tabs and in each tab examples for using the commands are shown. The different tabs and their operations are shown below:

- **Simple:** Here, simple mathematical computations like addition of two columns, subtraction of two columns, and scalar operations are listed.
- **Statistical:** Here, simple statistical operations like standard deviation and mean of columns are listed.
- **String:** Here, string matching operations and cancatenation of strings are listed.
- **Math:** Here, mathematical operations on columns like logarithm, exponent, etc are listed.
- **Condition:** Here, the if-then-else conditions on column operations are listed.
- **Count:** Here, count operations on each row that satisfy a certain condition ate listed.
- **Parameter Symbols:** Here, the way to use parameter symbols in the formula are given.

Examples of formulae appear on the user interface itself. Some caveats must be kept in mind while constructing formulae.

- Use * and + for "and" and "or" respectively.
- Remember to put braces while using and/or, so write $(d[0] > 5) * (d[0] < 8)$ instead of $(d[0] > 5 * d[0] < 8)$.

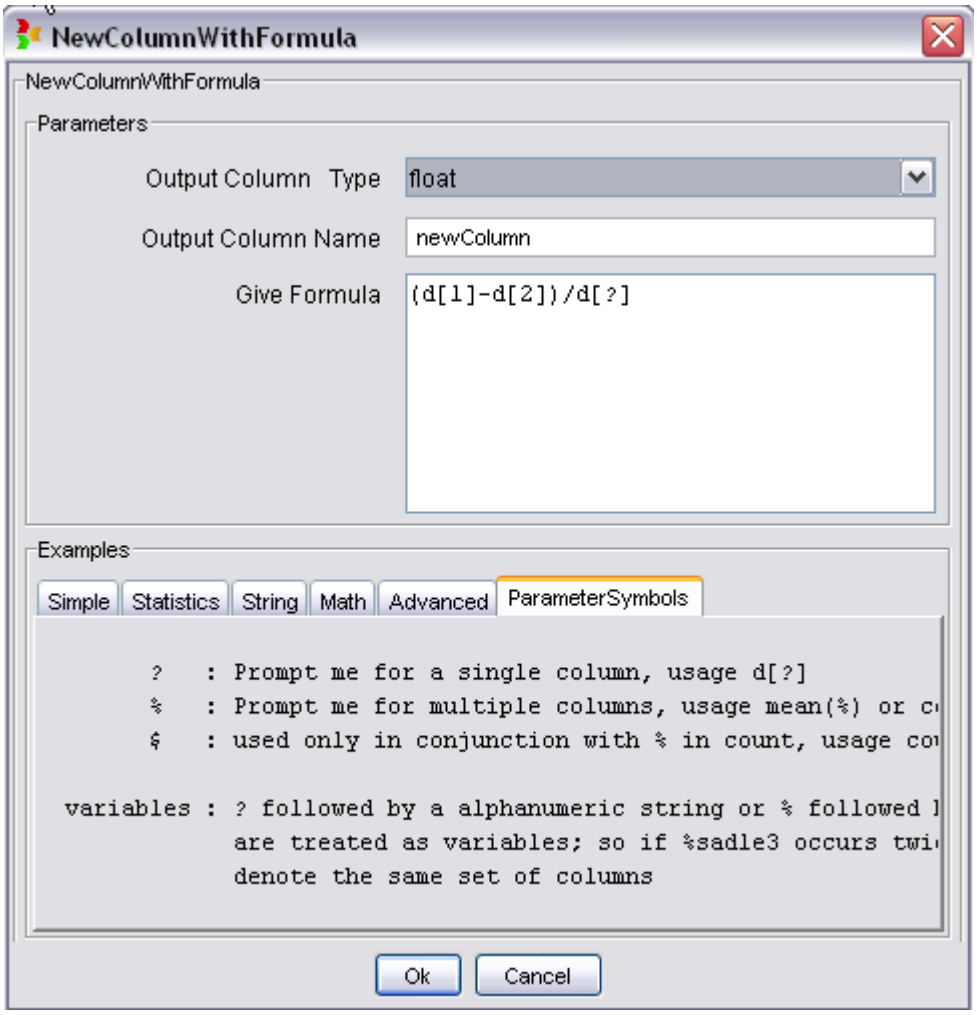


Figure 4.5: Create New Column by Formula

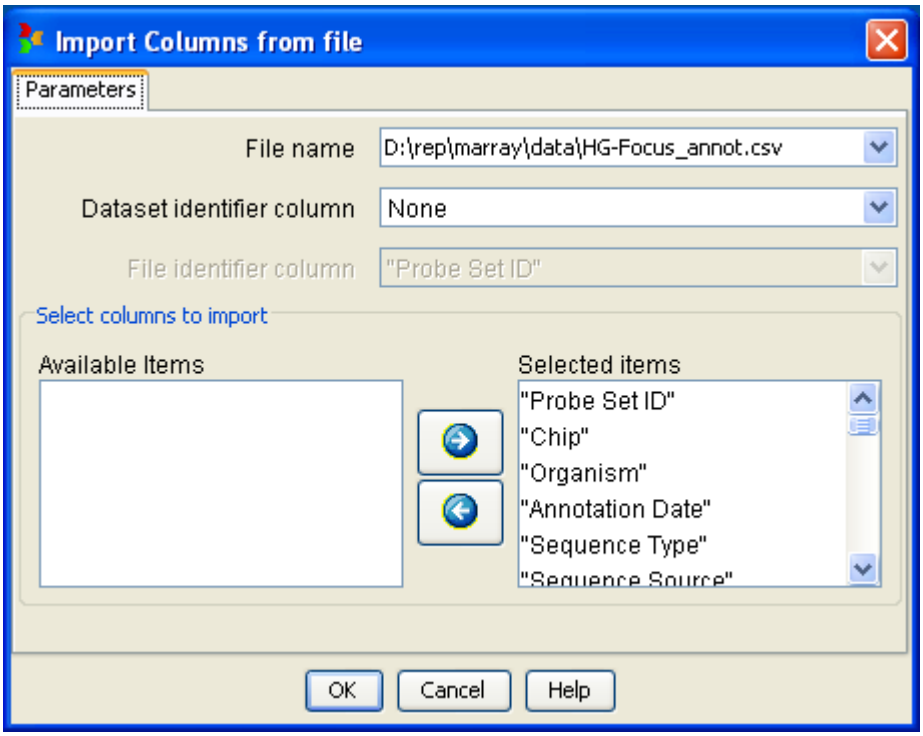


Figure 4.6: Import Columns from File

Remove Columns Use *Remove Columns* to remove selected columns from the dataset.

Import Columns

Use the *Column Import* option to import columns from a file into the dataset. This will pop up an Import Column Dialog. Browse and choose a file from which to import columns. This should be a structured comma separated (.csv) or tab separated file (.tsv or .txt). Lines beginning with "**##**" are considered as comment lines and ignored The first non-comment line is taken as the column header.

You can use a column to match and import data from the file, based on the values in the column. If an identifier column is marked on the dataset, this is chosen as the default Identifier column here. If an Identifier column in the dataset is chosen, you should choose a corresponding Identifier column in the file. If no Identifier column is chosen columns will be imported based on the row index.

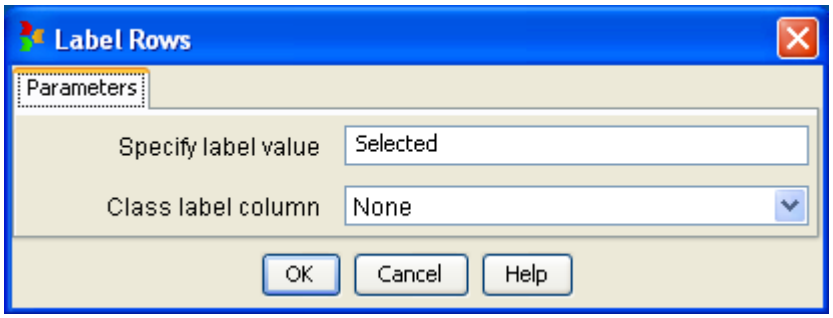


Figure 4.7: Label Rows

Choose the columns from the file to import and click OK. This will import the chosen columns into the current dataset.

4.1.2 Row Commands

Label Selected Rows: Selected rows can be labeled with specified label value. You can choose to add column to dataset and fill it with a label for the selected rows, else you can also update the values in any categorical coulnm of the dataset with a specified label for the selected rows. This feature is useful if certain coulmns need to be labeled for any downstream analysis.

4.1.3 Create Subset Dataset

You can create a subset dataset in the same project containing certain rows of the dataset. Subset dataset can be created from the selected rows; without the selected rows, or by removing all rows that contain missing values. This will create a subset dataset with the chosen parameters, as a child dataset in the project.

Create Subset from Selection: If certain rows or columns of the dataset are selected, this function will create a subset of the selected rows and columns. It will ask for a name for the child dataset and create a child dataset with the specified name. Note that all marked columns will be available in all the subset datasets in addition to the selected columns.

Create Subset by Removing selected Rows: This will create a subset dataset without the selected rows. It will ask for a name for the child dataset and create a child dataset with the specified name. Note that all marked columns will be available in all the subset datasets in addition to the selected columns.

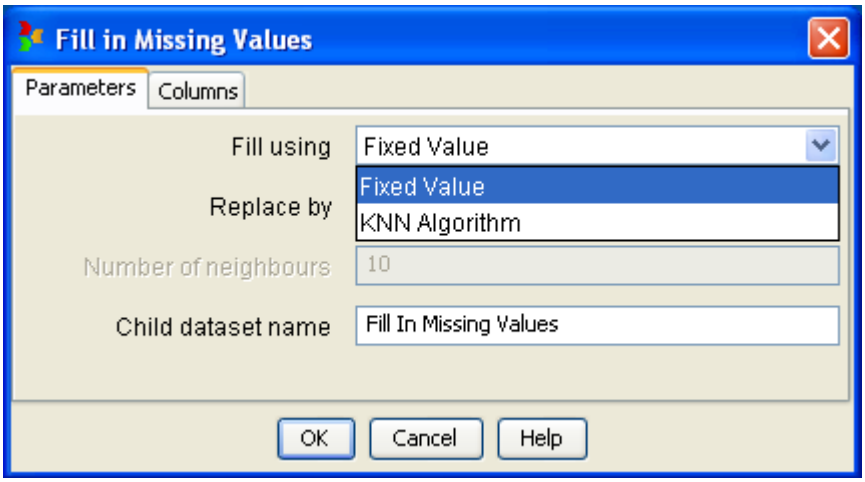


Figure 4.8: Setting Missing Values

Create a Subset by Removing Rows with Missing Values: Many algorithms do not run with missing values in the dataset. You may also want to remove all the rows with missing values from the dataset for further downstream analysis. Choosing this option will remove all rows with missing values from the dataset and create a child dataset with no missing values. It will ask for a name for the child dataset and create a child dataset with the specified name.

4.1.4 Transpose

Use this operation to create a spreadsheet in which rows become columns and columns become rows. If an Identifier column is marked then the values in this Identifier column will become the column names in the new dataset. If the Identifier column contains duplicate values, a number is appended to each duplicate value to make the column name unique. If no Identifier column is marked, then default column names will be added to the new dataset. Also, the column headers in the original dataset is transposed and is marked as the Identifier column in the new dataset. Note that the Transpose operation ignores all categorical columns in the dataset.

Set Missing Values

Several algorithms in **ArrayAssist** will not work if there are missing values in the data. A missing value will be marked as *N/A* in the spreadsheet. Use this operation to set missing values. These can be set to either a fixed constant value or by using the K-Nearest Neighbours (KNN) algorithm. This will replace all the missing values in the dataset with the value. The KNN algorithm finds the nearest neighbours to each missing value based on the values in other rows of the dataset and computes a value based on the K-nearest neighbours. If the particular value is missing in the K nearest neighbours and the algorithm is unable to impute a value for the mssing values, then the particular row will be removed from the child dataset. Also if more than 50 percent of the values in the rows are mssing, then the whole row is removed from the dataset. The dialog will ask for a name for the child dataset and create a child dataset with the specified name. After completing the algorithm, the a summary message with the number of rows removed and the number of missing values replaced is displayed.

Chapter 5

Importing Affymetrix Data

There are three possible starting points for analyzing data from Affymetrix arrays:

- Start with CEL files containing raw probe intensity data for each array.
- Start with CHP files for each experiment containing MAS5/PLIER output.
- Start with a tab separated text file containing MAS5 output for all arrays rolled into one file.

ArrayAssist provides extremely simplified interfaces to import CEL and CHP files via *File⟶New Affymetrix Expression Project⟶New Affymetrix project*. In particular, starting with CEL files is recommended for reasons described below. *File⟶Open* can be used to import and analyze tab or comma separated text files.

5.1 Key Advantages of CEL/CDF files

Affymetrix arrays have certain special probe characteristics. Each probeset has several associated *probe pairs*, with each probe pair comprising a Perfect Match and a Mismatch probe. Further, since probes are grown in-situ and packed densely, background correction cannot be performed by taking intensities in spot neighborhoods. Several specialized algorithms have emerged to handle these peculiarities; each of these has its own method for background subtraction, normalization, and probe summarization (i.e., averaging multiple probe values within a probeset into a single expression value). These algorithms include:

- The *RMA* algorithm due to Irazarry et al. [1, 2, 3].
- The *MAS5* algorithm, provided by Affymetrix [4].
- The *PLIER* algorithm due to Hubbell [5].
- The *dChip* algorithm due to Li and Wong [6].
- The *GCRMA* algorithm due to Wu et al. [7].

Comparative analysis of these algorithms on benchmark spike-in datasets has been performed by several researchers. The benchmark data used are the Affymetrix Latin Square series [8] and the GeneLogic spike-in and dilution studies [19]. Results of this comparative analysis have been published in [1, 2]. See [10] for a more exhaustive comparison. These studies clearly indicate that PLIER, RMA, DChip and GCRMA are all much superior to MAS5. These new algorithms can only be run starting with CEL files. **ArrayAssist** implements all of these algorithms thus providing researchers with a single unified platform for analysis.

5.2 Creating New Affymetrix Expression Project

Use the following command to import CEL/CHP files into **ArrayAssist**.

File⟶New Affymetrix Expression Project

This will launch a project wizard to take you through the steps for creating a new affymetrix expression project.

NOTE: Affymetrix CEL and CHP files are available in two formats, the Affymetrix GeneChip Command Console compliant data file (AGCC) files; and Extreme Data Access compliant data (GCOS XDA) files. **ArrayAssist** 5.1 uses the recently released Affymetrix Fusion SDKs that supports both AGCC and XDA format CEL and CHP files. However the older Affymetrix GDAC SDKs are also avaliable in **ArrayAssist**. By default, **ArrayAssist** uses the GDAC SDKs. The Fusion SDKs can be used by changing the defult settings in *Tools ⟶Options ⟶Affymetrix Probe-Level Analysis ⟶Fusion*

5.2.1 Selecting CEL/CHP Files

The first step in creating the project is to provide a project name and project folder. Click Next and select CEL or CHP files of interest. It is recommended that files not be mixed up, i.e., either only CEL files are chosen or only CHP files are chosen. To select files, click on the *Choose File(s)* button, navigate to the appropriate folder and select the files of interest. Use Left-Click to select the first file, Ctrl-Left-Click to select subsequent files, and Shift-Left-Click for a contiguous set of files. Once the files are selected, click on *Open* to load the files into the project. If you wish to select files from multiple directories or multiple contiguous chunks of files from the same directory, you can repeat the above exercise multiple times, each time adding one chunk of files to the selection window. You can remove already chosen files by first selecting them (using Left-Click , Ctrl-Left-Click and Shift-Left-Click , as above) and then clicking on the *Remove Files* button. After you have chosen the right files, hit the *Finish* button. If the library files for the corrseponding chip is available, the chips will be validated and the project will be loaded into **ArrayAssist**.

Finally, note that on Windows systems, you can choose to select CEL/CHP files directly from GCOS instead of the local file system by clicking on the *Load from GCOS* option. For more information, see Section on Importing Files from GCOS.

5.2.2 Getting Chip Information Packages

To import CEL and CHP files, you will need the *Chip Information Package* for your chip of interest. This package is a compact zip file containing probe layout information derived from the CDF file, probe affinity information pre-generated for running GCRMA, as well as gene annotation information derived from the NetAffx comma separated annotation file. If the *Chip Information Package* is not found, you will be prompted with a message asking you to download the required package. You can fetch this file using *Tools*$\longrightarrow$*Updates Data Library*$\longrightarrow$*From Web or From File* and then selecting the relevant package from the list of packages available.

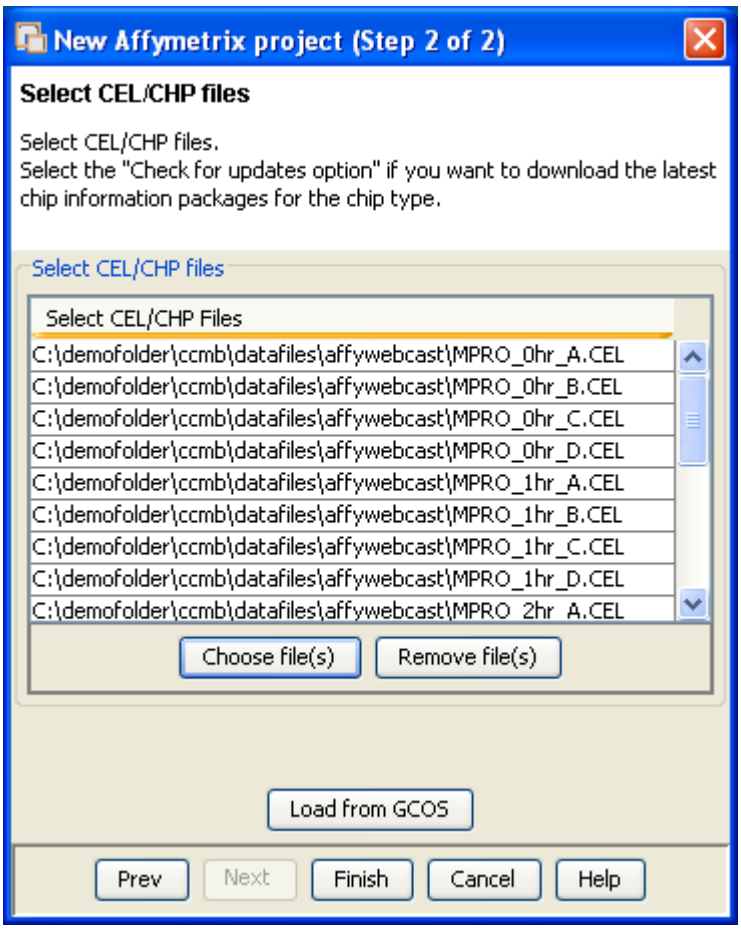


Figure 5.1: Choose CEL or CHP Files

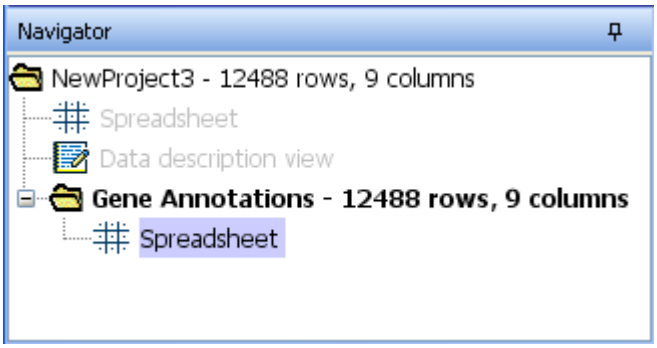


Figure 5.2: The Navigator at the Start of the Affymetrix Workflow

NOTE: **Chip Information Packages** could change every quarter as new gene annotations are released on NetAffx by Affymetrix. These will be put up on the **ArrayAssist** update server. **ArrayAssist** will directly keep track of the latest version available on **ArrayAssist** update server. When **ArrayAssist** launches, it will check the version available on the local machine with the version on the server. If a newer version has been deployed on the server, then, on starting, **ArrayAssist** will launch the update utility with the specific libraries check and marked for update.
Each project stores the generation date of the *Chip Information Package*. If newer libraries are available in the tool, when the project is opened, you will be prompted with a dialog asking you whether you want to refresh the annotations. Clicking on *OK* will update all the annotations columns in the project. You can also refresh the annotations after the project is loaded from the *Refresh Annotations* link in the workflow.

5.3 Running the Affymetrix Workflow

When the new Affymetrix project is created after proceeding through the above *File⟶Import Affymetrix Files⟶New Affymetrix project* wizard, **ArrayAssist** with open a new project with the following views:

The Data Description View: This view shows a list of CEL/CHP files imported in the panel on the left. The panel on the right has two tabs: File Header and Data. The *File Header* tab shows the file header containing some statistics for the file selected on the left panel. The *Data* tab shows the actual values in the selected file.

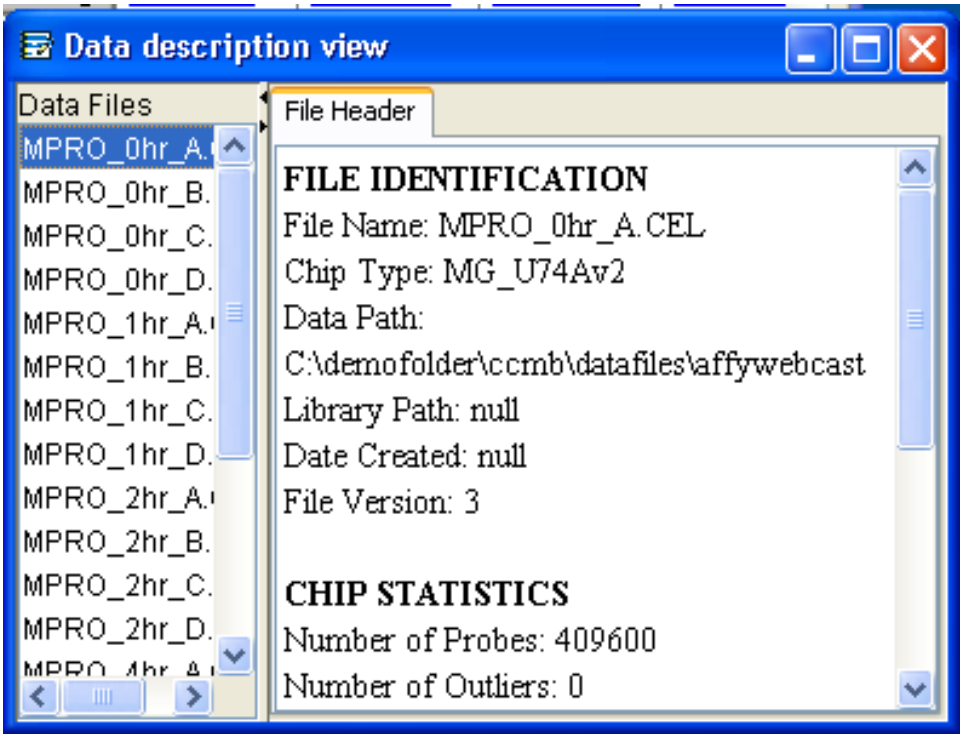


Figure 5.3: The Data Description View

The Spreadsheet: This is the Master dataset of the project. Initially, its contents will be the same as that of the Gene Annotations dataset. As the project is analyzed further, new derived columns, e.g., those obtained by running summarization algorithms, will be added to this master dataset. If you need to take a text export of all the derived columns, use the Right-Click *Export As⟶Text* option on this master dataset.

The Gene Annotations Dataset: Gene Annotations from NetAffx incorporated into the Chip Information Package are automatically extracted and displayed in the Gene Annotations dataset. Only a subset of the annotations available are imported by default to conserve space. The columns imported by default can be customized in *Tools⟶Options⟶Affymetrix Annotation Columns*. See Section on Fetching Gene Annotations from Web Sources for further details on using this dataset.

The ExpressionStat Dataset: This dataset is created only when importing CHP files and contains the signal values extracted from each of the CHP files. Gene Annotation columns can be brought into this dataset using Right-Click *Properties ⟶Columns*. Note that *ExpressionStat* refers to the name of the summarization algorithm used to create the CHP file as indicated in the Data Description view above (Affymetrix refers to the MAS5 algorithm as the ExpressionStat algorithm, CHP files generated using PLIER will lead to a Plier dataset).

The Absolute Calls Dataset: This dataset is also created only when importing CHP files and contains the absolute calls with corresponding p-values extracted from the CHP file along with two special columns showing the number of Present and Absent calls for each probeset. Gene Annotation columns can be brought into this dataset using Right-Click *Properties ⟶Columns*.

You are now ready to run the Affymetrix Workflow. The Affymetrix Workflow Browser contains all typical steps used in the analysis of Affymetrix microarray data. The very first step is providing Experiment Grouping. For more details, see Section on Project Setup. The remaining steps in the Workflow Browser are described below in detail. These steps will output various datasets and views and the following note will be useful in exploring these views.



Figure 5.4: The Affymetrix Workflow Browser

NOTE: Most datasets and views in **ArrayAssist** are *lassoed*, i.e., selecting one or more rows/columns/points will highlight the corresponding rows/columns/points in all other datasets and views. In addition, if you select probesets from any dataset or view, signal values and gene annotations for the selected probesets can be viewed using *View* ⟶*Lasso* (you may need to customize the columns visible on the Lasso view using Right-Click *Properties*).

5.3.1 Getting Started

Clicking on this link will take you to the appropriate chapter in the online manual giving details of loading expression files into **ArrayAssist**, the Affymetrix workflow, the method of analysis, the details of the algorithms used and the interpretation of results.

5.3.2 Project Setup

Experiment Grouping. Click on the Project Setup⟶Experiment Grouping to fill in details of your experimental design.

The *Experiment Grouping* view which comes up will initially just have the CEL/CHP file names. The task of grouping will involve providing more columns to this view containing *Experiment Factor* and *Experiment Grouping* information. A Control vs. Treatment type experiment will have a single factor comprising 2 groups, Control and Treatment. A more complicated Two-Way experiment could feature two experiment factors, genotype and dosage, with genotype having transgenic and non-transgenic groups, and dosage having 5, 10, and 50mg groups. Adding, removing and editing Experiment Factors and associated groups can be performed using the icons described below.

Reading Factor and Grouping Information from Files. Click on the Read Factors/Groups from File icon icon to read in all the Experiment Factor and Grouping information from a tab or comma separated text file. The file should contain a column containing CEL/CHP file names; in addition, it should have one column per factor containing the grouping information for that factor. Here is an example tab separated file. The result of reading this tab file in is the new columns corresponding to each factor in the Experiment Grouping view.

```
#comments
#comments
```

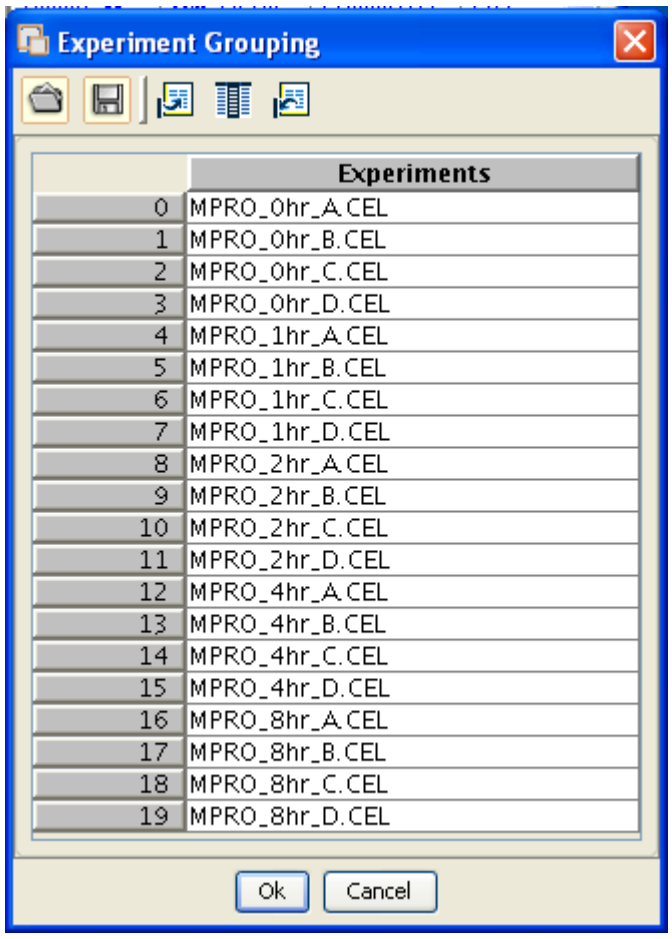


Figure 5.5: The Experiment Grouping Step in the Affymetrix Workflow Browser

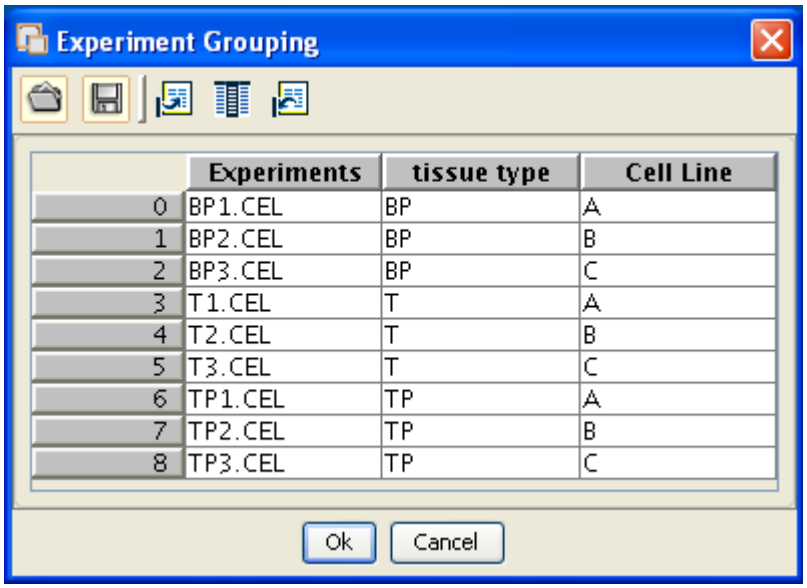


Figure 5.6: The Experiment Grouping View With Two Factors

```
filename genotype dosage
A1.CEL     NT        0
A2.CEL      T        0
A3.CEL     NT       20
A4.CEL      T       20
A5.CEL     NT       50
A6.CEL      T       50
```

Adding a New Experiment Factor. Click on the Add Experiment Factor icon to create a new Experiment Factor and give it a name when prompted. This will show the following view asking for grouping information corresponding to the experiment factor at hand. The CEL/CHP files shown in this view need to be grouped into groups comprising biological replicate arrays. To do this grouping, select a set of CEL/CHP files, then click on the *Group* button, and provide a name for the group. Selecting CEL/CHP files uses Left-Click , Ctrl-Left-Click , and Shift-Left-Click , as before.

Editing an Experiment Factor. Click on the Edit Experiment Factor icon to edit an Experiment Factor. This will pull up the same grouping interface described in the previous paragraph. The groups already set here

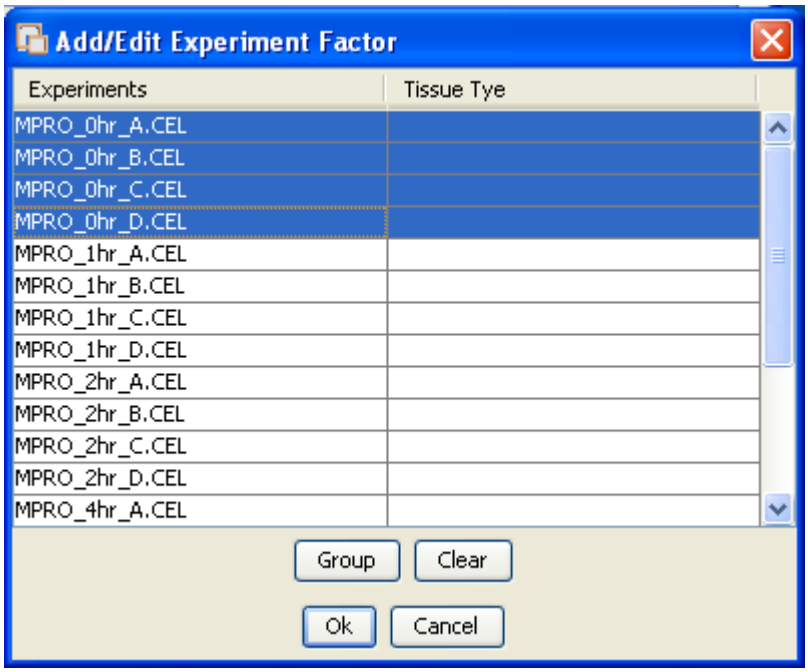


Figure 5.7: Specify Groups within an Experiment Factor

can be changed on this page.

Remove an Experiment Factor. Click on the Remove Experiment Factor  icon to remove an Experiment Factor.

5.3.3 Primary Analysis

The primary analysis of Affymetrix Expression Project consists of three steps, *Probe Level Analysis*, *Quality Control* and *Data Transformations*

Probe Level Analysis

You will need to run this step only if you imported CEL files; for CHP files, the ExpressionStat and AbsoluteCalls datasets represent the results of summarization, i.e., these are the *Summarized* datasets.

Probe Summarization for CEL files can be performed by clicking on the appropriate links in the Affymetrix Workflow browser. Click on Primary Analysis, probe Level Analysis. This will show the following options. Click on the desired summarization algorithm to run it.

- RMA
- MAS5
- PLIER
- LiWong or dChip
- GCRMA

Each of these algorithms will create a new *Summarized* dataset containing signal values on the *linear* scale (in contrast to previous versions of **ArrayAssist** which used the log scale). In addition, the MAS5 algorithm will also create an Absolute Calls dataset. This dataset will contain the absolute calls and corresponding p-values along with two special columns showing the number of Present and Absent calls for each probeset. To see a description of the columns in any dataset, use *Data*$\longrightarrow$*Properties*.

Note that you can run multiple algorithms within the same project. For instance, if you wish to run RMA but would still like to filter on absolute calls, then run RMA and then MAS5. Now, select the RMA summarized dataset in the navigator, and finally filter on calls using the link in the Workflow Browser described in Filter on Calls and Signals.

For more details on the above algorithms and configurable parameters, if any, see Section on Probe Summarization Algorithms

Quality Control

One you have a *Summarized* dataset, the next step would be to check for sample and quality. **ArrayAssist** provides the following workflow steps to do this.

NOTE: Remember to select a Summarized dataset on the navigator before running one of the following steps.

Hybridization Quality Plots Clicking on this link will output 3 types of sample and hybridization quality views:

The *Internal Controls* view depicts RNA sample quality by showing 3'/5' ratios for a set of specific probesets which include the actin and GAPDH probesets. The 3'/5' ratio is output for each such probeset and for each array. The ratios for actin and GAPDH should be no more than 3 (though for Drosophila, it should be less than 5). A ratio of more than 3 indicates degradation of RNA during the isolation process. Note that when invoked for a MAS5 summarized dataset, the Internal Controls view will also show absolute calls. A ratio greater than 3 is often overlooked if the call is A.

The *Poly-A Controls* view is used to monitor the entire target labeling process. Dap, lys, phe, thr, and trp are B. subtilis genes that have been modified by the addition of poly-A tails and then cloned into pBluescript vectors which contain T3 promoter sequences. Amplifying these poly-A controls with T3 RNA polymerase will yield sense RNAs, which can be spiked into a complex RNA sample, carried through the sample preparation process, and evaluated like internal control genes. The final concentrations of the controls, relative to the total RNA population, are: 1:100,000; 1:50,000; 1:25,000; 1:7,500, respectively. All of the Poly-A controls should be called Present with increasing Signal values in the order of lys, phe, thr, dap, trp. The Poly-A control view will show the signal value profiles of these transcripts (with signals averaged over the 3' and 5' probesets). There is one profile for each array, with the Legend at the bottom-right showing on mouseover which profile corresponds to which array. Often, it may be useful to view these profiles on the log-scale which can be done via Right-Click *Properties*. The Absolute Calls for these transcripts can be obtained from the Absolute Calls dataset obtained by running MAS5 summarization. Go to the Absolute Calls dataset, sort the Probeset Id column so that all the AFFX- probes appear together at the top, select rows corresponding to the above transcripts and then scroll right to the Number of Present Calls and Number of Absent Calls columns.

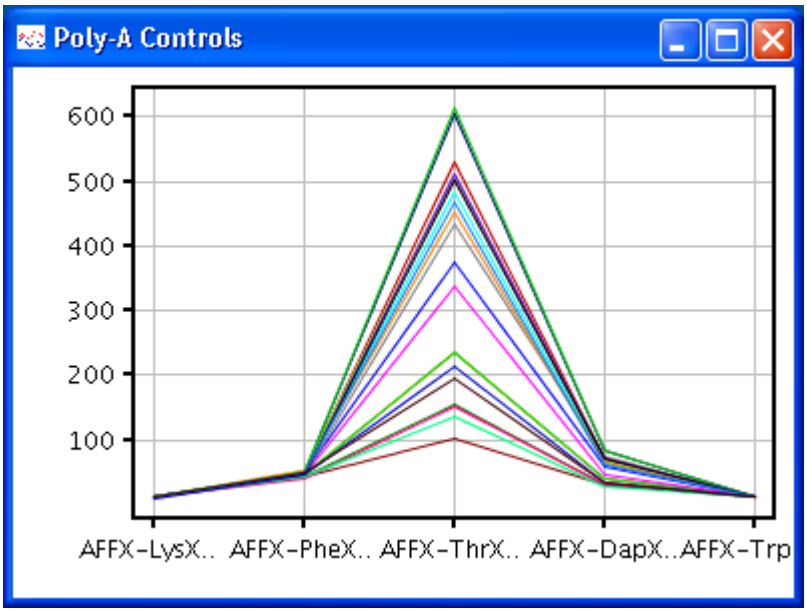


Figure 5.8: Poly-A Control Profiles

The *Hybridization Controls* view depicts the hybridization quality. Hybridization controls are composed of a mixture of biotin-labelled cRNA transcripts of bioB, bioC, bioD, and cre prepared in staggered concentrations (1.5, 5, 25, and 100pm respectively). This mixture is spiked-in into the hybridization cocktail. bioB is at the level of assay sensitivity and should be called Present at least 50% of the time. bioC, bioD and cre must be Present all of the time and must appear in increasing concentrations. The Hybridization Controls view shows the signal value profiles of these transcripts (only 3' probesets are taken). There is one profile for each array, with the Legend at the bottom-right showing which profile corresponds to which array. Often, it may be useful to view these profiles on the log-scale which can be done via Right-Click *Properties*. The Absolute Calls for these transcripts can be obtained from the Absolute Calls dataset obtained by running MAS5 summarization. To do this, go to the Absolute Calls dataset, sort the Probeset Id column so that all AFFX- probes appear together at the top, select rows corresponding to the above transcripts and then scroll right to the Number of Present Calls and Number of Absent Calls columns.

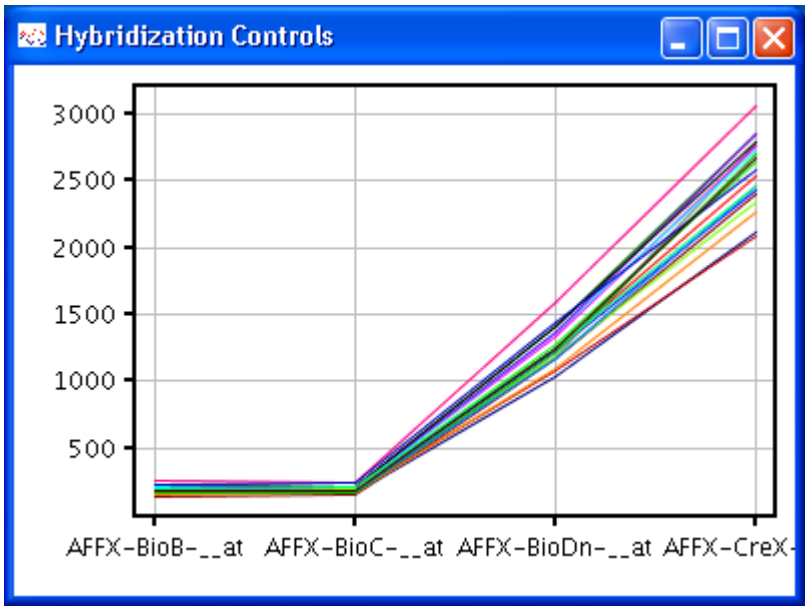


Figure 5.9: Hybridization Control Profiles

Data Quality Plots This step is for checking visual consistency across arrays, i.e., whether the data is well normalized or not. Clicking on this link will output a scatter plot, and a statistics view. The scatter plot will show the first two arrays; other arrays can be viewed by changing the X and Y axes using the drop-down list. The plots should produce approximately 45 degree plots for the arrays to be consistent. Sometime the scatter plots are better viewed on the log scale, which can be set via Right-Click *Properties*. The statistics plot shows distributions of signal values within each array, which should also be consistent across arrays.

Principal Component Analysis on Arrays. This link will perform principal component analysis on the arrays. It will show the standard PCA plots (see PCA for more details). The most relevant of these plots used to check data quality is the *PCA scores* plot, which shows one point per array and is colored by the Experiment Factors provided earlier in the *Experiment Grouping* view. This allows viewing of separations between groups of replicates. Ideally, replicates within a group should cluster together and separately from arrays in other groups. The *PCA scores* plot can be color customized via Right-Click *Properties*. All the Experiment Factors should

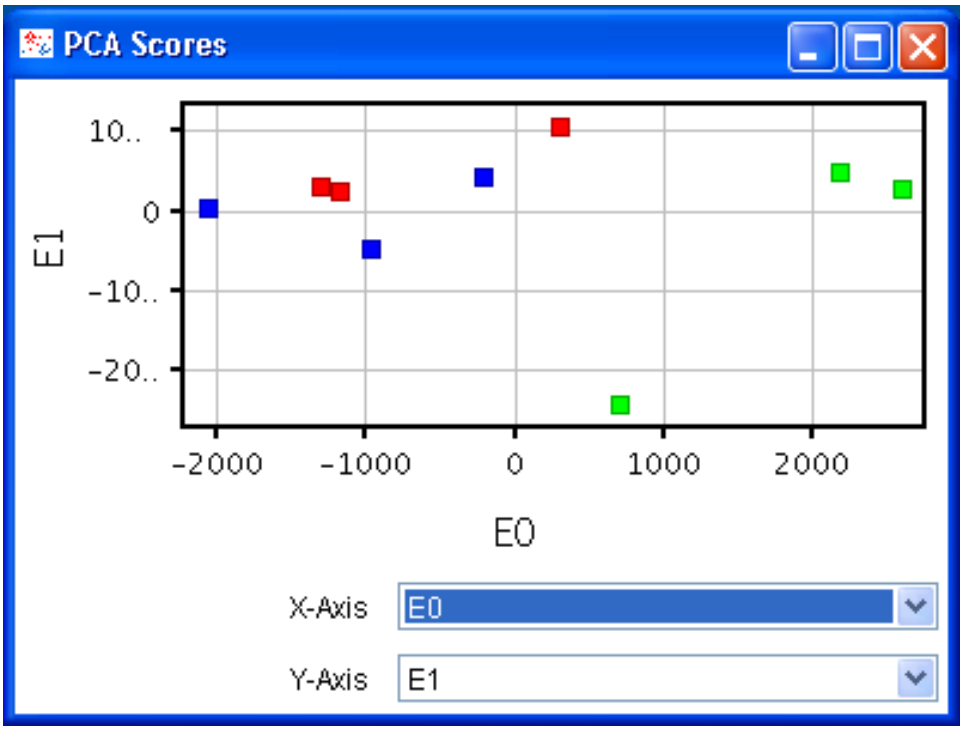


Figure 5.10: PCA Scores Showing Replicate Groups Separated

occur here, along with the Principal Components E0, E1 etc. The PCA Scores view is *lassoed*, i.e., selecting one or more points on this plot will highlight the corresponding columns (i.e., arrays) in all the datasets and views. Further details on running PCA appear in Section on PCA.

Correlation Plots. This link will perform correlation analysis across arrays. It finds the correlation coefficient for each pair of arrays and then displays these in two forms, one in textual form as a correlation table view, and other in visual form as a heatmap. The heatmap is colorable by Experiment Factor information via Right-Click *Properties*. The intensity levels in the heatmap can also be customized here. The text view itself can be exported via Right-Click *Export as Text*. Note that unlike most views in **ArrayAssist**, the correlation views are not *lassoed*, i.e., selecting one or more rows/columns here will not highlight the corresponding rows/columns in all the other datasets and views.

Sometimes it is useful to cluster the arrays based on correlation. To do this, export the correlation text view as text, then open it via *File*$\longrightarrow$*Open*,

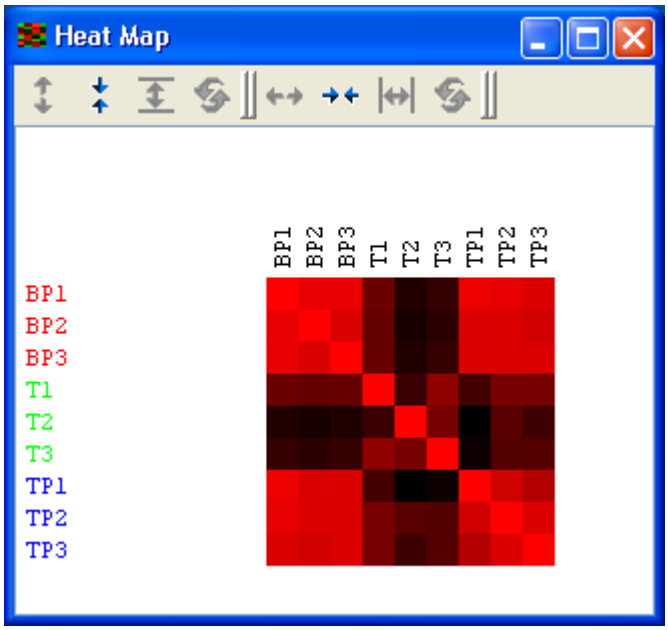


Figure 5.11: Correlation HeatMap Showing Replicate Groups Separated

and then use *Cluster*⟶*Hier* to cluster. Row labels on the resulting dendrogram can then be colored based on Experiment Factors using Right-Click *Properties*.

5.3.4 CHP/RPT/MAGE-ML Writing

Once summarization is done, the summarized data and results can be exported in various formats. All summarized data can be exported as CHP files and in MAGE-ML format. RPT report files can also be generated from any summarized dataset. However, only CHP files of MAS5 summarized data can be exported into GCOS.

Write CHP File. This will write CHP files with the summarized values for each of the CEL files in the project. This will operate only on a summarized dataset. The CHP files will be written into an appropriate folder in the default project directory. The CHP files can later be used to create a New Affymetrix Project.

This will also launch a view of the CHP files giving the File Identification, Chip Statistics and the Algorithm Details.

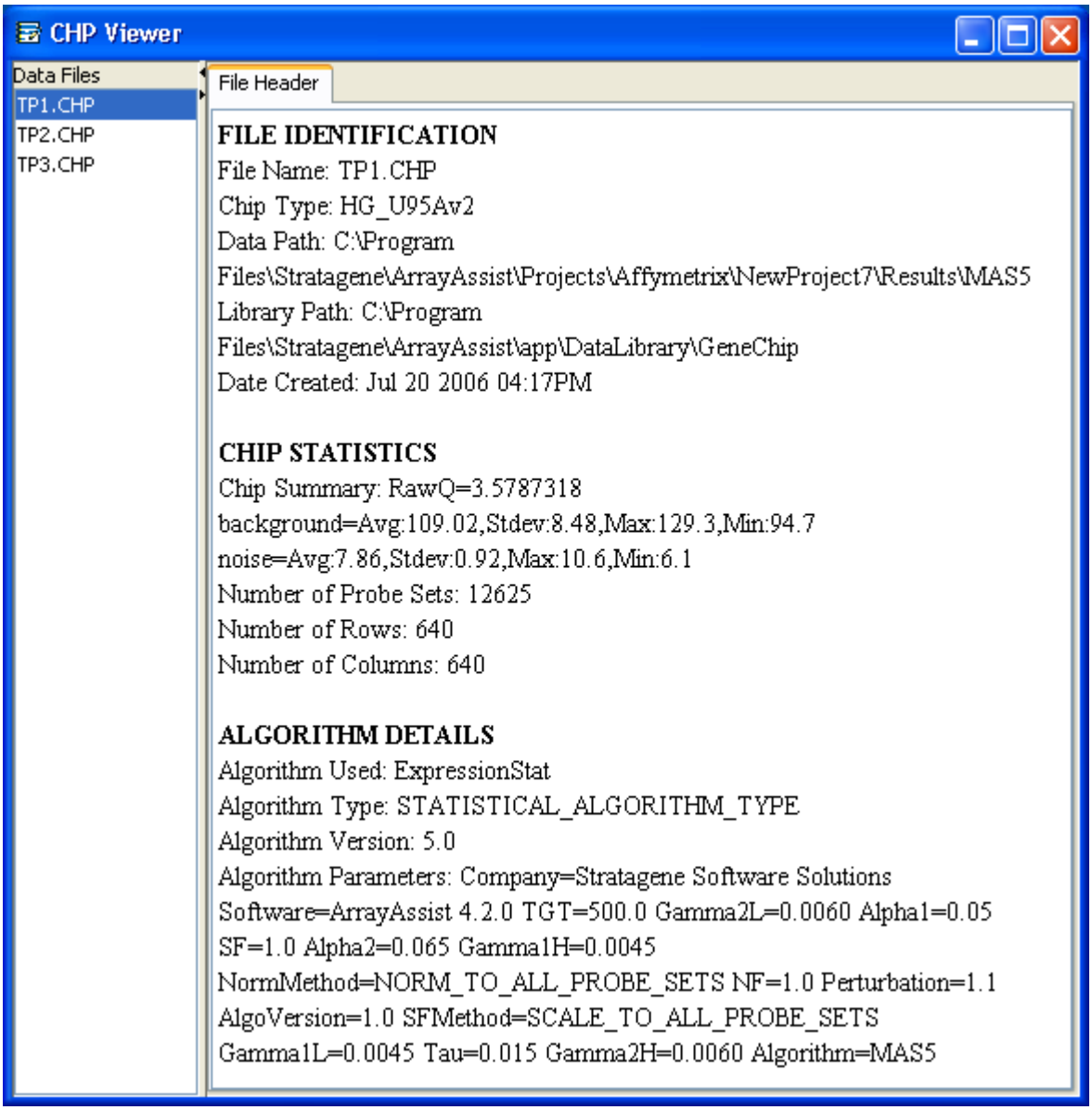


Figure 5.12: CHP Viewer

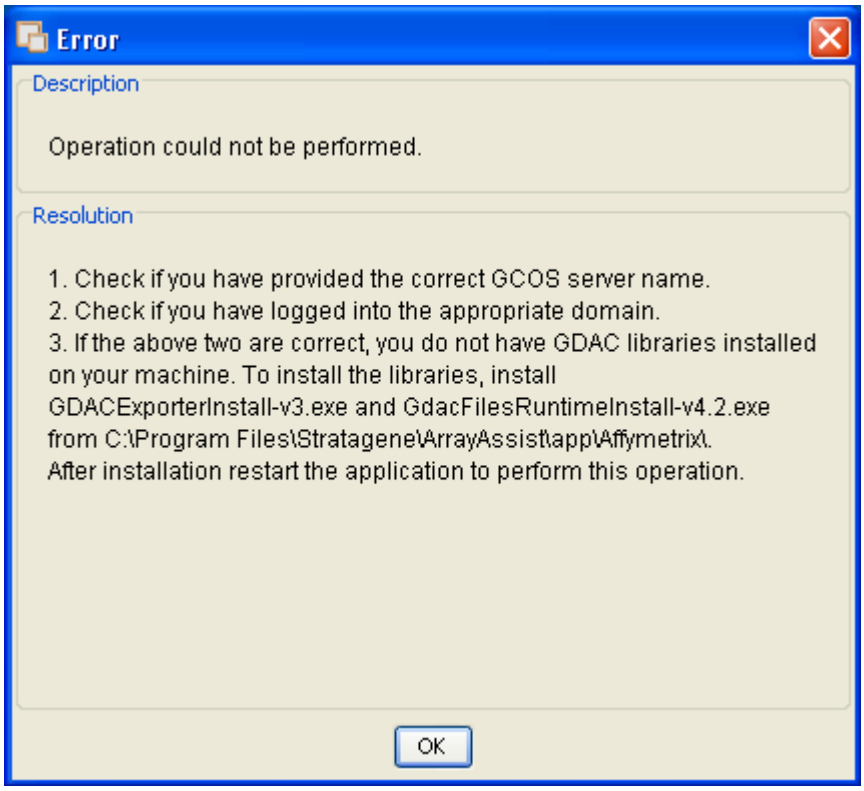

Error
Description
Operation could not be performed.
Resolution
1. Check if you have provided the correct GCOS server name.
2. Check if you have logged into the appropriate domain.
3. If the above two are correct, you do not have GDAC libraries installed on your machine. To install the libraries, install GDACExporterInstall-v3.exe and GdacFilesRuntimeInstall-v4.2.exe from C:\Program Files\Stratagene\ArrayAssist\app\Affymetrix\.
After installation restart the application to perform this operation.
OK

Figure 5.13: GCOS Error

Write CHP files to GCOS. To write CHP file to GCOS you will need some additional libraries provided by Affymetrix. If you have the GCOS Client installed on your machine, these libraries will already be present on your machine. If you are trying to access a GCOS server on your network, you will be prompted to install these libraries on your machine. Follow the on-screen instructions to install these libraries the installers for which are packed with **ArrayAssist**.

Once you have the required libraries, you can write the CHP files to the GCOS client / server system. If you want to write to the GCOS Server, you will have to be logged into the GCOS Server domain and have the appropriate permissions.

Provide the server name when prompted. This server name is the name of your local machine if it runs the GCOS workstation, or the name of the machine running the GCOS server, if you are running a remote server. To

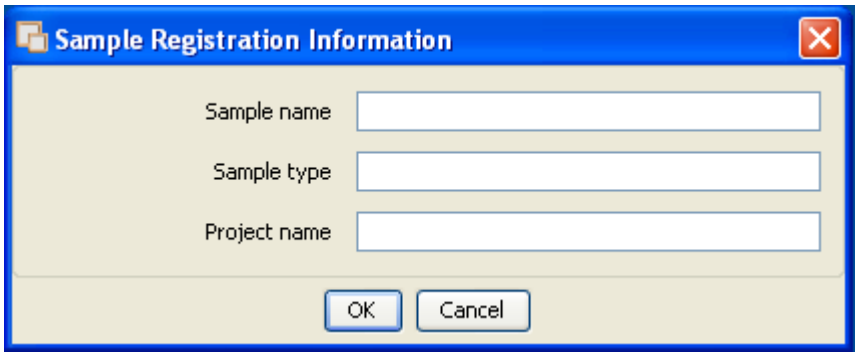


Figure 5.14: Register Sample in GCOS

find the machine name, right click on My Computer, go to Properties and then to the tab Network Identification or Computer Name. (Note that you will have to give the GCOS Server Name and not the ipaddress).

Writing to GCOS will register the CHP files with the GCOS system and copy the files into the GCOS system. This operation can only be performed on a MAS5 summarized dataset. The CHP files can then be used to create a New Affymetrix Project. You will be asked for the name of the project and other details of the project when you write the CHP file into GCOS. Note that the library files for the CHP must be installed on the GCOS client / server.

The GCOS Server Name can also be provided in the *Tools* $\longrightarrow$*Options* dialog. (Note that you will have to provide the GCOS Server Name and not the ipaddress)

Write RPT Files. Clicking on this link will create a report and write the RPT file into an appropriate folder in the default project directory. The RPT report will be also be displayed in an report view on the desktop.

MAGE-ML Writer. To write RPT files you will need some additional libraries provided by Affymetrix. If you do not have these libraries, when you click on this link, you will be prompted to install these libraries. Follow the on-screen instructions to install these libraries the installers for which are packed with **ArrayAssist**.

This will create a MAGE-ML output of all the CEL and CHP files in the project. One MAGE-ML file will be written for each CEL file in the project along with a text file containing the data.

This will create a MAGE-ML output of all the summarized CEL files.

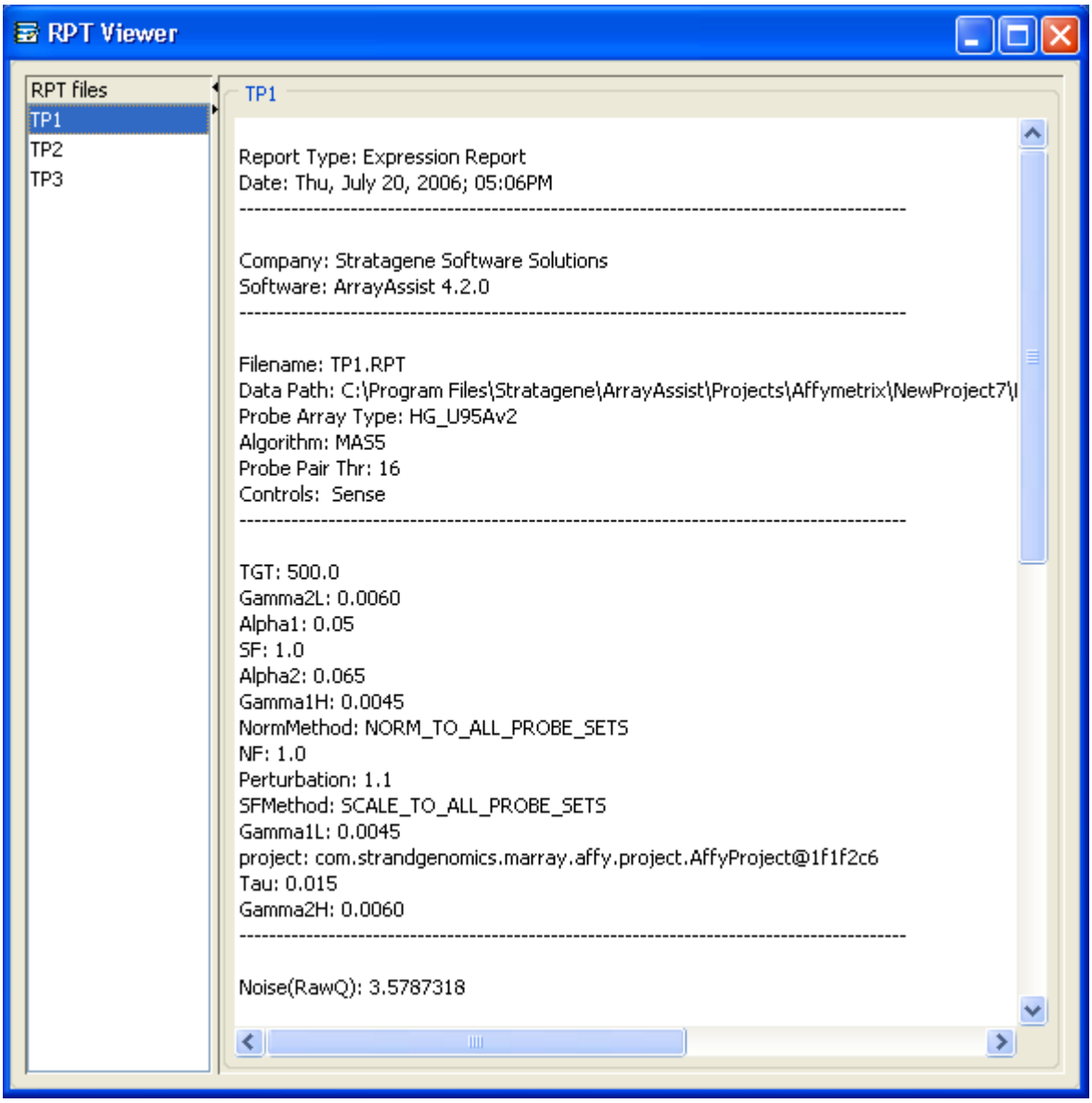


Figure 5.15: RPT View

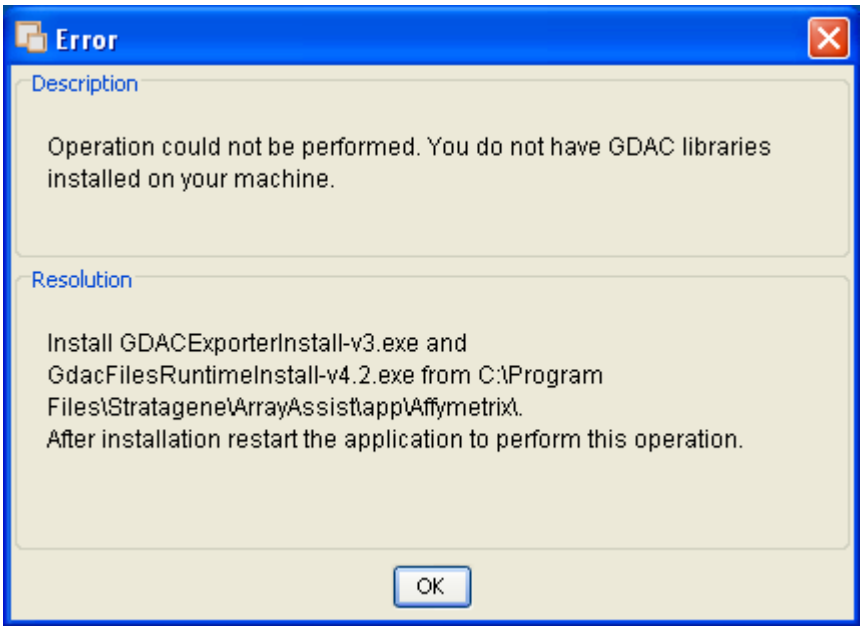

Error
Description
Operation could not be performed. You do not have GDAC libraries installed on your machine.
Resolution
Install GDACExporterInstall-v3.exe and GdacFilesRuntimeInstall-v4.2.exe from C:\Program Files\Stratagene\ArrayAssist\app\Affymetrix\.
After installation restart the application to perform this operation.
OK

Figure 5.16: MAGE-ML Error

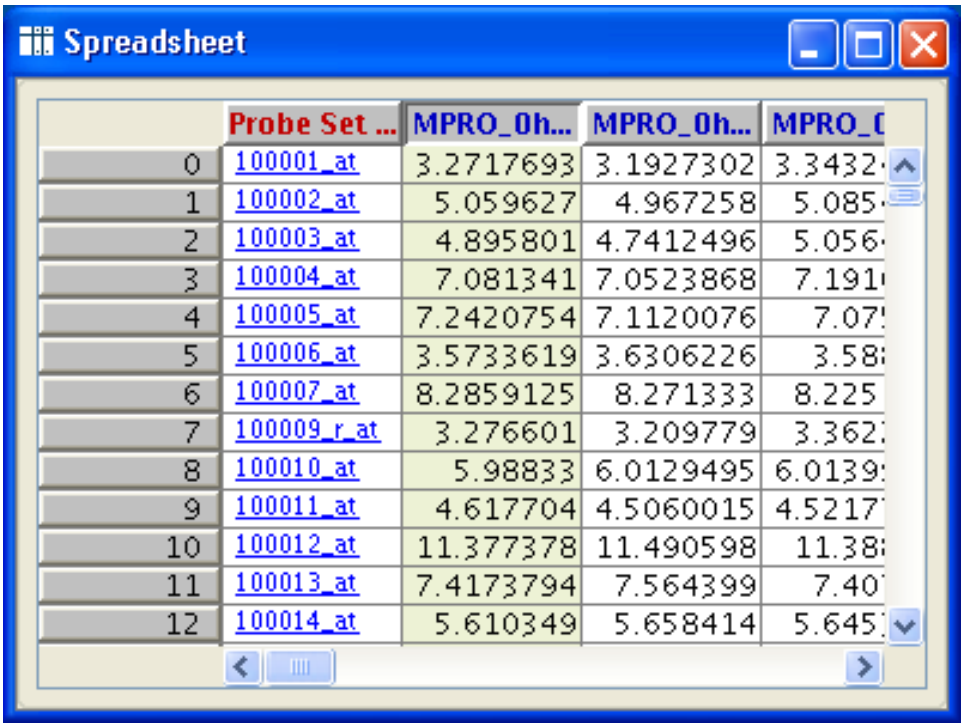


Figure 5.17: New Child Dataset Obtained by Log-Transformation

One MAGE-ML file will be written for each CEL file in the project along with a text file containing the data.

5.3.5 Data Transformations

Once data is summarized and quality has been checked for, the next step is to perform various transformations. The list of transformations available in the workflow browser is described below. Each transformation will produce a new child dataset in the navigator. Each of these datasets will have access to gene annotation information which can be brought into the respective spreadsheets using Right-Click *Properties* $\longrightarrow$*Columns*. Also, rows and columns in each of these datasets will be *lassoed* with the rows and columns, respectively, in all the other datasets. Selecting a row/column in one dataset with highlight it in all the other datasets and open views, making it easy to track objects across datasets and views.

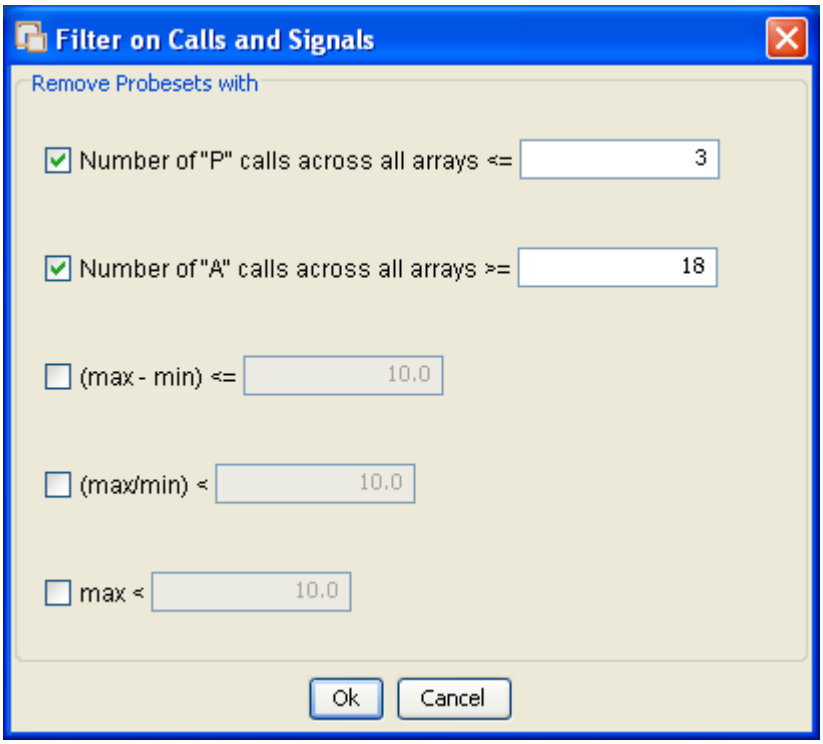


Figure 5.18: Filter on Calls and Signals Dialog

NOTE: Data transformation will often require you to select a specific dataset in the navigator. For example, Log-Transformation will require selecting a Summarization dataset containing signal values (obtained via one of the summarization algorithms or via the import of CHP files). Appropriate messages will be displayed if the right dataset is not selected in the Navigator.

Filter on Calls and Signals. Use this step to filter genes based on Absolute Calls and Signal values. To perform this step you must have an Absolute Call dataset already generated and visible in the navigator. To generate this dataset, either run the MAS5 algorithm or import CHP files generated using the MAS5 algorithm. Once you have an absolute call dataset, *select the summarized dataset* you are interesting in filtering and run this transformation. It comes up with a dialog.

This dialog supports filtering based on the options listed below. You can choose any subset of these by ticking on the appropriate checkboxes. If

multiple checkboxes are checked, then probesets which satisfy ANY of the corresponding conditions are removed.

- Remove Probesets with Number of "P" (Present) calls across all arrays $\leq$ (at most) a specified amount. This will create a new dataset with only those probesets which have more Present calls than the threshold. Signal values in this new dataset will be derived from the selected summarized dataset.

- Remove Probesets with Number of "A" (Absent) calls across all arrays $\geq$ (at least) a specified amount. This will create a new dataset with only those probesets which have fewer Absent calls than the threshold. Signal values in this new dataset will be derived from the selected summarized dataset.

- Remove Probesets with (max-min) signal value $\leq$ (at most) a specified amount. This will create a new dataset with only those probesets for which the difference between the maximum signal value over all arrays and the minimal signal value over all arrays is at least the threshold, i.e., there is substantial variation across arrays.

- Remove Probesets with (max/min) signal value $<$ a specified amount. This will create a new dataset with only those probesets for which the ratio of the maximum signal value over all arrays to the minimal signal value over all arrays is at least the threshold, i.e., there is substantial variation across arrays.

- Remove Probesets with max signal value $<$ a specified amount. This will create a new dataset with only those probesets for which the maximum signal value over all arrays is more than the threshold.

Note that the log transformation should be performed only after this step.

Variance Stabilization. Use this step to add a fixed quantity (16 or 32) to all linear scale signal values. This is often performed to suppress noise at log signal values, e.g., as shown in the pre- and post- variance stabilization scatter plots generated by PLIER summarization. Log transformation should be performed only after variance stabilization.

Logarithm Transformation. Use this step to convert linear scale data to logscale, where logs are taken to base 2. This step is *necessary* before performing statistics, baseline transformations and computing sample averages; these transformations will work only on log-transformed summarized datasets.

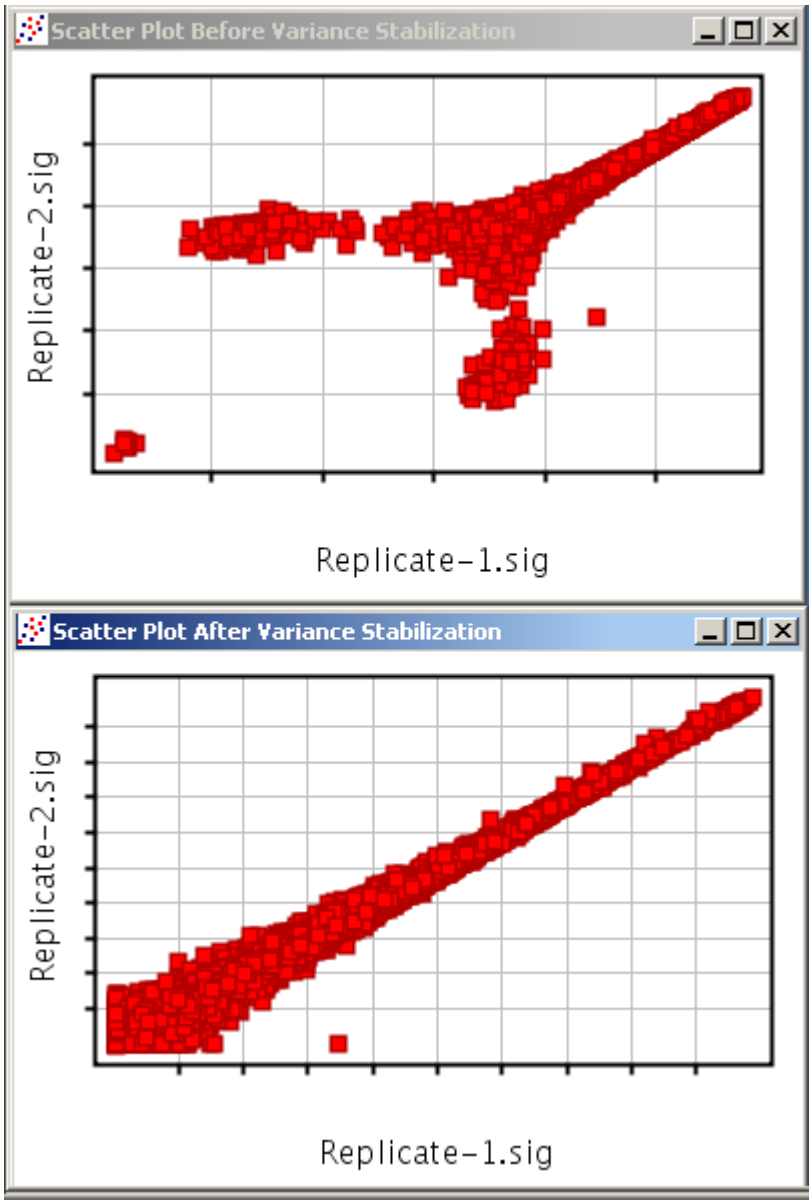


Figure 5.19: Variance Stabilization

Baseline Transformation. This step only works on log-transformed summarized datasets and produces *log-ratios* from log-scale signals. The ratios are taken relative to the average value in a specified experiment group called the *Baseline* group.

Recall that experiment factors and groups were provided earlier as in Section on Project Setup. One of these groups of replicate arrays will serve as the baseline. Next, the log-scale signal values of each probeset will be averaged over all arrays in the baseline group. This amount will be subtracted from each log-scale signal value for this probeset in the log-transformed summarized dataset. This transform is useful primarily for viewing (e.g., in a heatmap, colors in the baseline group are subdued and all others reflect a color relative to this baseline group, in particular, positive and negative log ratios relative to this group are well differentiated).

To run this transformation, you will need to specify the baseline group. To this effect, **ArrayAssist** will ask you first to choose an experiment factor amongst those provided prior to generating signal values. Next, it will ask you to choose the baseline group from within the groups for this experiment factor.

Compute Sample Averages. This step only works on log-transformed summarized datasets and averages arrays within the same replicate groups to obtain a new set of averaged arrays. Recall that experiment factors and groups were provided earlier as in Section on Project Setup. To run this transformation, you will need to specify the experiment factor(s) and group(s) over which averaging needs to be performed. For instance, you may choose one experiment factor and all or a few groups corresponding to this factor; the averages within each of the chosen groups will be computed. If you choose multiple experiment factors, say factor A with groups AX and AY and factor B with groups BX and BY, then averages will be computed within the 4 groups, AX/BX, AX/BY, AY/BX, and AY/BY. The result of running this transformation will be a new dataset containing the group averages. By using the up/down arrow keys on the dialog shown below, the order of groups in the output dataset can be customized.

5.3.6 Data Exploration

Data in datasets within an Affymetrix project can be visualized via the views in the *Views* menu as well as the view icons on the toolbar. Each view allows various customizations via the Right-Click *Properties* menu. Some views which operate on specific columns or subsets of columns will use the column selection in the currently active dataset by default. To select columns in a

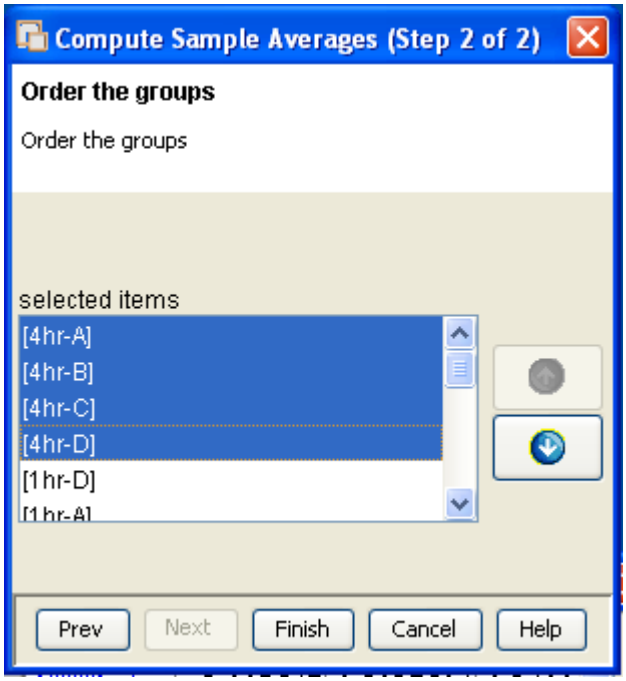


Figure 5.20: Reorder Groups for Viewing

dataset use Left-Click , Ctrl-Left-Click , Shift-Left-Click on the body of the column (and not on the header). For more details on the various views and their properties, see the chapter on Data Visualization.

The Affymeytrix Workflow browser currently provides the following additional viewing options.

Scatter Plot. This will launch a scatter plot of the logarithm transform signal columns of the current dataset. Various pairs of columns can be chosen for viewing.

MVA Plot. This will launch an MVA plot of the signal columns of the dataset. If the data has been normalized, the MVA plot will show the scatter along the zero line.

Profile Plot by Group. This view option allows viewing of profiles of probesets across arrays comprising specific experiment factors and groups of interest. Recall that experiment factors and groups were provided earlier as in Section on Project Setup. To obtain this plot, you will need to specify the experiment factor(s) and group(s) over which averaging needs to be performed. For instance, you may choose one experiment factor and all or a few groups corresponding to this factor; you can then also use the up/down arrows to specify the order in which the various groups will appear on the plot. A profile plot with the arrays comprising these groups, in the right order, will be presented.

Histogram. This will launch a histogram of the individual signal columns of the dataset. This view is helpful to view the distribution of the signal values for each experiment.

Matrix Plot. This will launch a matrix plot of the signal columns of the dataset. The Matrix plot will show by default the first three arrays. More arrays can be viewed using the Right-Click $\longrightarrow$*Properties* $\longrightarrow$*Rendering* tab and changing the number of rows and columns. (Remember to press *Enter* after putting in each value.)

5.3.7 Significance Analysis

ArrayAssist provides a battery of statistical tests including T-Tests, Mann-Whitney Tests, Multi-Way ANOVAs and One-Way Repeated Measures tests. Clicking on the *Significance Analysis Wizard* will launch the full wizard which will guide you through the various testing choices. Details of these choices appear in Section on The Differential Expression Analysis Wizard, along with detailed usage descriptions. For convenience, a few commonly

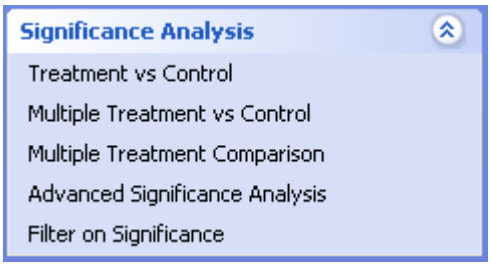


Figure 5.21: Significance Analysis Steps in the Affymetrix Workflow

used tests are encapsulated in the Affymetrix Workflow as single click links; these are described below.

The Treatment vs Control: This link will function only if the Experiment Grouping view has only one factor, which comprises two groups. You will be prompted for which of the two groups is to be considered as the Control group. A standard T-Test is then performed between Treatment and Control groups. P-values, Fold Changes, Directions of Regulation (up/down), and Group Averages are derived for each probeset in this process. In addition, P-values corrected for multiple testing are also derived using the Benjamini-Hochberg FDR method (see Differential Expression Analysis for details).

The Multiple Treatment vs Control: This link will function only if the Experiment Grouping view has only one factor, which comprises more than two groups. You will be prompted for which of the groups is to be considered as the Control group. Subsequently, each non-Control group will be T-Tested against the Control group. P-values, Fold Changes, Directions of Regulation (up/down), and Group Averages are derived for each probeset in each T-Test. In addition, P-values corrected for multiple testing are also derived using the Benjamini-Hochberg FDR method (see Differential Expression Analysis for details).

Multiple Treatments Comparison: This link will function only if the Experiment Grouping view has only one factor, which comprises more than two groups. A One-Way ANOVA will be performed on all these groups. P-values and Group Averages are derived for each probeset in this process. In addition, P-values corrected for multiple testing are also derived using the Benjamini-Hochberg FDR method (see Differential Expression Analysis for details).

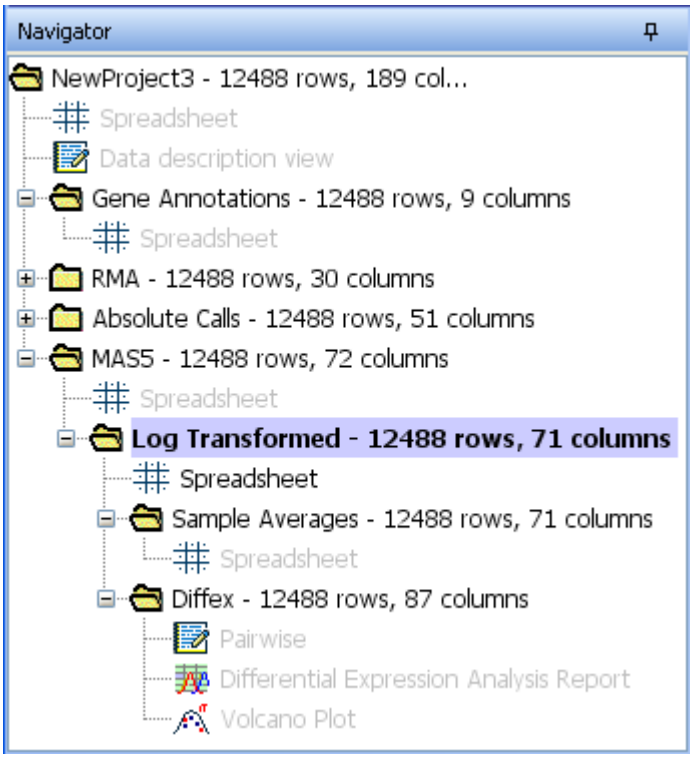


Figure 5.22: Navigator Snapshot Showing Significance Analysis Views

NOTE: Significance Analysis between and Treatment and Control group will output a table and volcano plots of Treatment Vs Control. all computations of fold change and direction of regulation will always be computed as Treatment/Control. In general if the Significance tests is done choose X vs Y, the fold change will always be given as X/Y

Results of Significance Analysis are presented in views and datasets described below. All of these appear under the *Diffex* node in the navigator as shown below.

The Statistics Output Dataset. This dataset contains the p-values and fold-changes (and other auxiliary information), generated by Significance Analysis.

The Differential Expression Analysis Report. This report shows the test type and the method used for multiple testing correction of p-values.

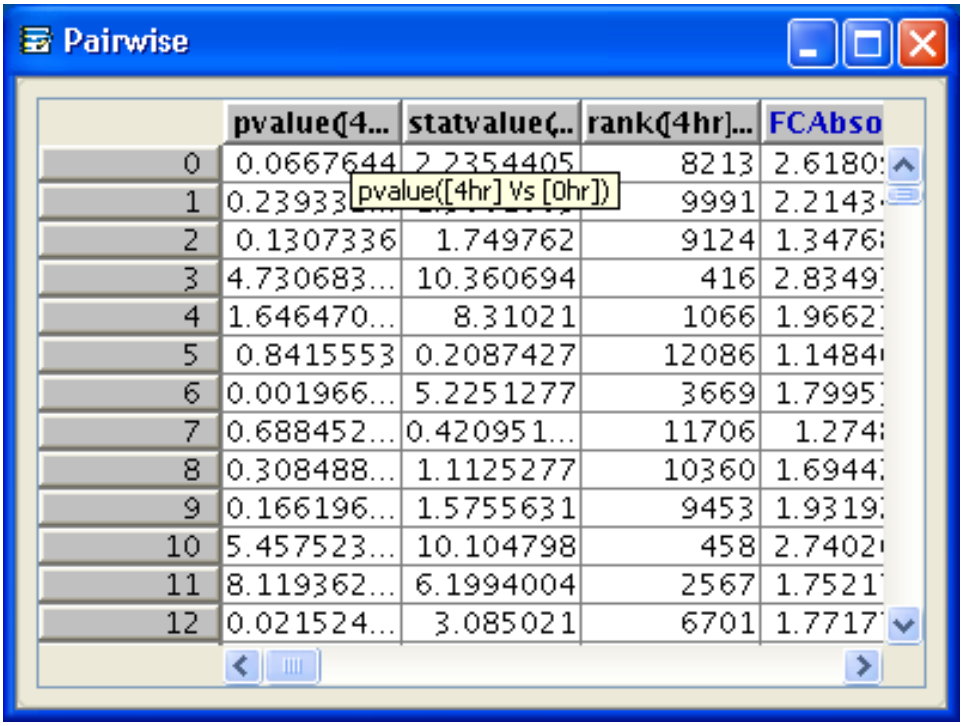


Figure 5.23: Statistics Output Dataset for a T-Test

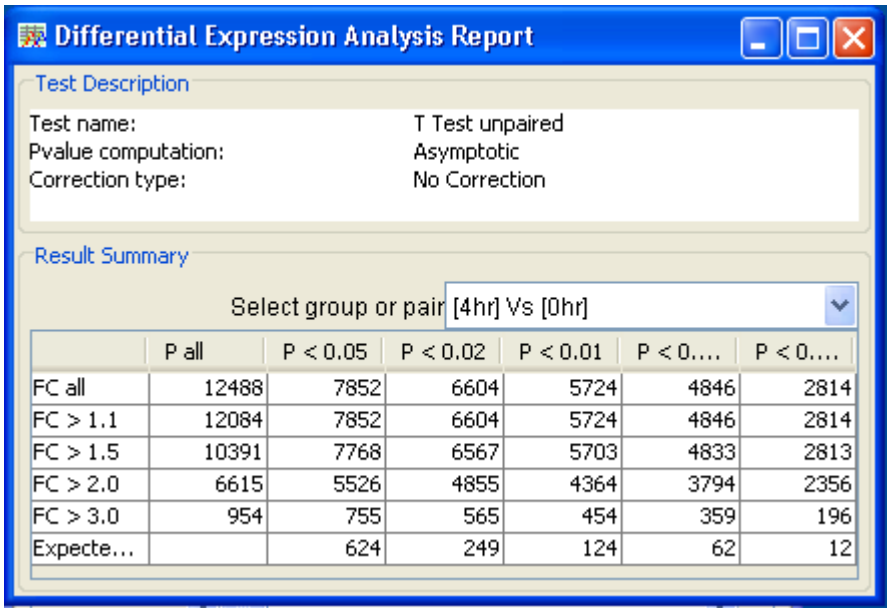


Figure 5.24: Differential Analysis Report

In addition, it shows the distribution of genes across p-values and fold-changes in a tabular form. For T-Tests, each table cell shows the number of genes which satisfy the corresponding p-value and fold-change cutoffs. For ANOVAs, each table cell shows the number of genes which satisfy the corresponding fold-change cutoff only. For multiple T-Tests, the report view will present a drop down box which can be used to pick the appropriate T-Test. Clicking on a cell in these tables will select and lasso the corresponding genes in all the views. Finally, note that the last row in the table shows some *Expected by Chance* numbers. These are the number of genes expected by pure chance at each p-value cut-off. The aim of this feature is to aid in setting the right p-value cutoff. This cut-off should be chosen so that the number of gene expected by chance is much lower then the actual number of genes found (see Differential Expression Analysis for details).

The Volcano Plot. This plot shows the log of p-value scatter-plotted against the log of fold-change. Probesets with large fold-change and low p-value are easily identifiable on this view. The properties of this view can be customized using Right-Click *Properties*.

Filtering on p-values and Fold Changes. There are two ways to filter.

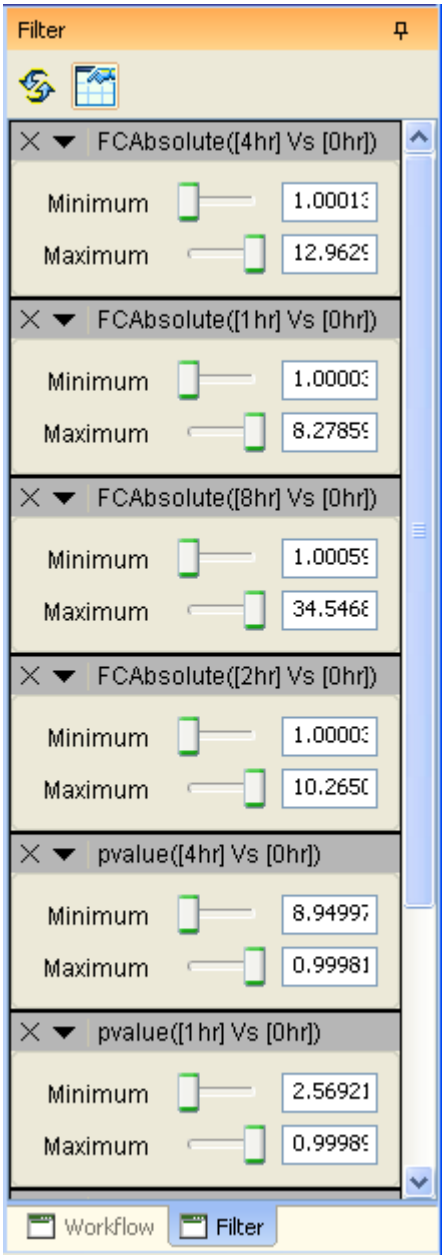


Figure 5.25: Filtering

The first and simpler option uses the Filter on Significance Link in the workflow browser. Fill in cut-offs for p-value, fold-change and regulation (up, down or both). Conditions on the various groups shown in this dialog are combined via an "and", i.e., all of the specified cut-offs must be satisfied.

The second method is as follows. Go to the Statistics Output dataset in the navigator. Then, in the *Filter*, click on the Properties  icon and move the appropriate columns (p-value, fold-change etc.) from the left to the right. Sliders corresponding to these columns will now appear on the filter as shown in the figure below. Setting the appropriate values on these sliders (either via the sliders themselves or via the associated text boxes; remember to press the enter key after modifying text in a text box) will filter away the relevant genes from ALL datasets. Now, go to any dataset of interest, select all rows in this dataset using Left-Click , Ctrl-Left-Click , Shift-Left-Click on the row headers, and then use *Data*⟶*Create Subset*⟶*with Selection* to create a child dataset containing the genes of interest. You can then reset the filter using the Reset Filter  icon.

For a more complex scenario, consider situations where you do two separate statistical tests and want to identify genes with a p-value less than say 0.05 in one experiment and p-value greater than .1 in the other. You can run the above filtering steps on each of the two statistics output datasets as follows. Start with the first Statistics Output dataset, use the Filter to restrict all datasets to the relevant genes, and then use *Data*⟶*Row Commands*⟶*Label Selected Rows* to add a label identifying these genes. Then repeat this with the second Statistics Output dataset, adding a second label this time. Now use the filter on these label columns to restrict all datasets to the required genes.

5.3.8 Clustering

The only clustering link available from the workflow browser is the K-Means which clusters the signal columns into 10 clusters. To run another algorithm or to change parameters, use the *Cluster* menu. See Section on Clustering for more information.

NOTE: The default clustering in the workflow link runs the k-means cluster and will automatically use the signal columns in the dataset to run the clustring algorithm.
When clustering is called from the menu bar, a clustring parameters dialog will pop-up. By default all the continuous columns in the active dataset will be selected in the clustering algorithm. You will have to go to the columns tab in the clustering parameters dialog, select the appropriate signal columns in the dataset and run the clustering algorithm. Alternatively, you can select the appropriate signal columns in the spreadsheet and then call the clustering algorithm. Selected columns will be used for clustering.

5.3.9 Save Probeset Lists

After running significance analysis and clustering, when certain probes of interest have been identified, you may want to save the probes as a separate probeset list. These could be used with other probeset to draw Venn Diagrams and visualize unions and intersections. Create a selection of Probesets of interest and click on the Create Probeset List from Selection. This will pop-up a dialog with the name of the Gene list and the identifier for the Gene list. By default, the Affymetrix ProbeSet Id will be chosen as an identifier. You can change the Identifier to any of the marked columns in the dataset from the drop-down list provided.

5.3.10 Import annotations

Click on the Import Annotations link to import additional annotations into the dataset. All the annotations that are available with the NetAffx annotation are available with the library files. However, by default only a few important annotations are loaded when the project is created. To load additional annotations from NetAffx, click on this link. This will bring up a dialog with all available annotation columns. Choose the required columns and move them to the Selected Items list and click OK. This will import the selected columns into the dataset.

5.3.11 Discovery Steps

As mentioned earlier in Section , gene annotations from NetAffx are automatically imported at the time of new project creation. The columns to be imported from NetAffx can be specified in the project creation wizard. These columns appear in the The Gene Annotations Dataset. Like all

datasets, this dataset also supports selection, filtering, subsetting and variety of other operations (see Create Subset Dataset). Some further specific operations available from the workflow browser are described below.

Fetching Gene Annotations from Web Sources. You can fetch annotations for selected genes from various public web sources. Select the genes of interest from any dataset or view, then choose the gene annotations dataset on the Navigator and click on this link. Select the public source of your interest, and indicate the input gene identifier you wish to start with (unigene, genbank accession etc) and the information you need to fetch (gene name, alias etc). The information fetched will be updated in the gene annotations dataset or appended in some cases when the column fetched is not already there in the dataset. Note that the input identifiers used need to be marked (see Section Marking Annotation Columns), i.e., identified as unigene, genbank accession etc. To mark a column, use *Data* $\longrightarrow$*Data Properties* and set the appropriate marks using the dropdown list provided for each column. Alternatively, the Annotation wizard has an option to mark columns. For more details on the public sites accessible and of the input and output identifiers, see the chapter on Annotating Genes.

• Note that several of the columns in the Gene Annotation dataset are hyperlinked, for instance the Probeset Id is linked to the Affymetrix NetAffx page, Gene Ontology accession is linked to the AMIGO page etc. For a list of these hyperlinks, see *File*$\longrightarrow$*Configuration*$\longrightarrow$*AffyURL*. These hyperlinks can be edited here.

Gene Ontology Browser. You can view *Gene Ontology* terms for the genes of interest in the Gene Ontology Browser invokable from this link. This browser offers several queries, a few of which are detailed below. See Section on GO Browser for a more complete description.

• To view GO Terms for genes of interest and to identify enriched GO Terms, select genes of interest from any view and then click on the

Find Go Terms with Significance  icon.

Next move to the *Matched Tree view*. Here you will see all Gene Ontology terms associated with at least one of the genes along with their associated enrichment p-value (see Section on GO Computation for details on how this is computed). You can navigate through this tree to identify Go Terms of interest.

• A tabular view of the p-values can also be obtained by clicking on the

P-Value Dataset  icon

icon. This will produce a table in which rows are the above visible GO terms, and the columns contain various statistics (i.e., enrichment p-value, the number of genes having a particular GO term in the entire array, the number of genes amongst those selected having a particular GO term etc.).

• Another tabular dataset can be obtained by clicking on the
GeneVsGo Dataset  icon
and providing a cut-off p-value. This dataset shows probesets along the rows and GO Terms which occur in at least one of these probesets along the columns, with each cell being 0 or 1 indicating the presence or absence of that GO term for that probeset. This view is best viewed as a HeatMap by selecting the relevant columns and launching the HeatMap view from the *View* menu.

• You can also begin with a GO term (select it in the Full Hierarchy tab, if necessary you can use the search function to locate the term), and then click on
Find All Genes with this Term  icon.
This will select all probesets having this particular GO term in all the views and datasets.

Your currently active dataset needs to contain a Gene Ontology Accession column and this must be marked as such a column via *Data* $\longrightarrow$*Properties*. Each cell in this column should be a pipe separated list of GO terms, e.g., GO:0006118|GO:0005783|GO:0005792|GO:0016020.

Viewing Chromosomal Locations. Click on this link to view a scatter plot between Chromosome Number and Chromosome Start Location. Each probeset is depicted by a thin vertical line. Each chromosome is represented by a horizontal bar. Each probeset can be given a color as well. For instance, to color probesets by their fold changes or p-values, go to the Statistics output dataset in the Navigator and then launch the Chromosome Viewer. Use Right-Click *Properties* to color by the p-value or fold change columns.

Importing Gene Annotations from Files. If you have your own set of gene annotations which you wish to import, prepare these annotations as a tab or comma separated file with genes as rows and annotation fields (name, symbol, locuslink etc.) as columns. Then import this file by going to the gene annotations dataset and using *Data* $\longrightarrow$*Columns*$\longrightarrow$*Import Columns*. Provide the file name and the gene identifier to be used for synchronizing

columns in the file imported with columns in the gene annotations dataset. Next, mark each of the imported columns by setting the appropriate column mark in the *Data Properties* (appropriate marks include Unigene Id, Gene Name etc.). This will ensure two things: first, that these new columns are available from all child datasets, and second, that these columns are interpreted correctly by the annotation modules (web spidering, GO Browsing etc).

Note that there is a small problem in *importing annotations from NetAffx csv files* using the above method. This file has strings enclosed containing commas which spoil the comma separated structure. To parse this correctly, you will need to open this file in excel and save it as a tab separated txt file. Alternatively, use the **ArrayAssist** *File* $\longrightarrow$*Import Wizard* to import the file and then save it as a tab separated txt file; remember to use quotes as the text indicator in the import process. For large files, it is recommended that you take the first 100 lines, put it through the **ArrayAssist** *File* $\longrightarrow$*Import Wizard* and create a template. Now, use this template to import the whole file.

Creating Custom Links. You can cause entries in a particular column to be treated as hyperlinks by changing the column mark to URL in *Data* $\longrightarrow$*Data Properties*. Subsequently, clicking on an entry in this column (either in the spreadsheet or in the lasso) will open the corresponding link in an external browser. Note that the entries in this column must be hyperlinks (i.e., of the form http:// etc.).

In case you wish to create a new hyperlink column, use the *Data*$\longrightarrow$*Column* $\longrightarrow$*Append Columns By Formula* command to create an appropriate string column and then use *Data* $\longrightarrow$*Data Properties* to mark this column as a URL column. For more details on creating new columns with formulae, see Section on Create New Column using Formula.

5.3.12 Genome Browser

The Genome Browser can be invoked using this link. This browser allows viewing of several static prepackaged tracks. In addition, new tracks can be created based on currently open datasets. For more details on usage, see Section on The Genome Browser.

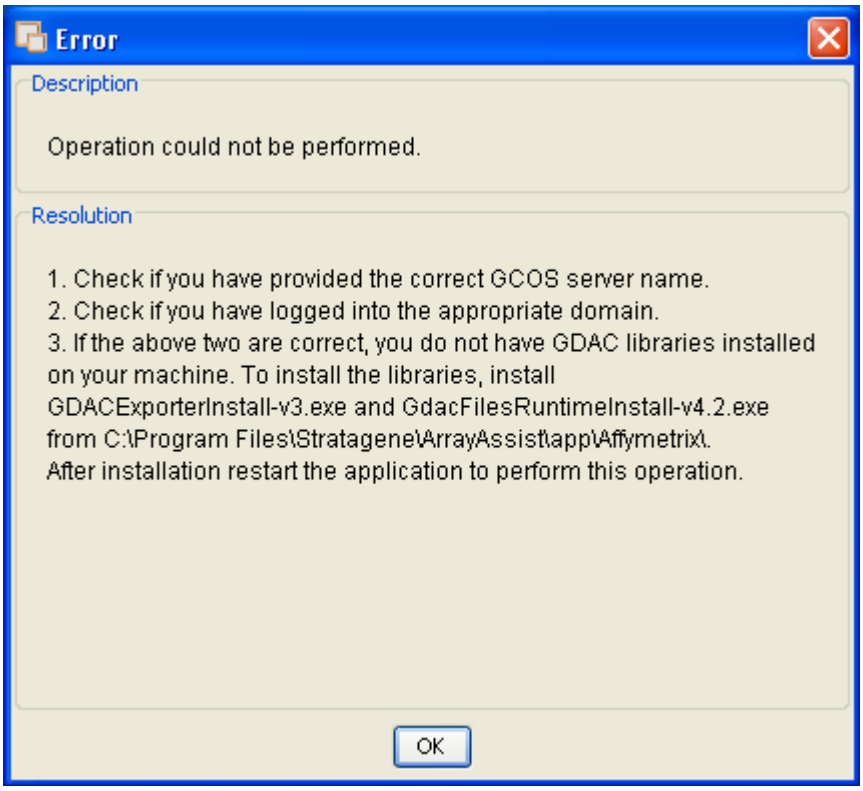

Error
Description
Operation could not be performed.
Resolution
1. Check if you have provided the correct GCOS server name.
2. Check if you have logged into the appropriate domain.
3. If the above two are correct, you do not have GDAC libraries installed on your machine. To install the libraries, install GDACExporterInstall-v3.exe and GdacFilesRuntimeInstall-v4.2.exe from C:\Program Files\Stratagene\ArrayAssist\app\Affymetrix\.
After installation restart the application to perform this operation.
OK

Figure 5.26: GCOS Error

5.4 Importing CEL/CHP Files from GCOS

ArrayAssist can read CEL and CHP files directly from the Affymetrix GCOS system, without having to export the files out of GCOS. You will need to have either a GCOS Client installed on your local machine or the GCOS server running on a remote machine on your LAN. To access files from GCOS you will need some additional libraries provided by Affymetrix. If you have the GCOS Client installed on your machine, these libraries will already be present on your machine. If you are trying to access a GCOS server on your network, you will be prompted to install these libraries on your machine. The installer for these libraries are packaged with **ArrayAssist**.

Once the libraries are available is installed, you will need to provide the GCOS server name in the *File* $\longrightarrow$*New Affymetrix Project* wizard. To import files from the server, you will have to be logged into the GCOS server

domain and you should have the appropriate permissions. Choose the *Load from GCOS* option and provide the server name when prompted. This server name is the name of your local machine if it runs the GCOS workstation, or the name of the machine running the GCOS server, if you are running a remote server. To find the machine name, right click on My Computer, go to Properties and then to the tab Network Identification or Computer Name. (Note that you will have to give the GCOS Server Name and not the ipaddress). After the name is given, there might be a substantial pause followed by the popping up of the GCOS filechooser, allowing selection of CEL/CDF files from within GCOS.

The GCOS Server Name can also be provided in the *Tools* $\longrightarrow$ *Options* dialog. (Note that you will have to provide the GCOS Server Name and not the ipaddress)

5.5 Technical Details

This section describes technical details of the various probe summarization algorithms, normalization using spike-in and housekeeping probesets, and computing absolute calls.

5.5.1 Probe Summarization Algorithms

Probe summarization algorithms perform the following 3 key tasks: Background Correction, Normalization, and Probe Summarization (i.e. conversion of probe level values to probeset expression values in a robust, i.e., outlier resistant manner. The order of the last two steps could differ for different probe summarization algorithms. For example, the RMA algorithm does normalization first, while MAS5 does normalization last. Further, the methods mentioned below fall into one of two classes – the PM based methods and the $PM - MM$ based methods. The $PM - MM$ based methods take $PM - MM$ as their measure of background corrected expression while the PM based measures use other techniques for background correction. MAS5, MAS4, and Li-Wong are $PM - MM$ based measures while RMA and **ArrayAssist** are PM based measures. For a comparative analysis of these methods, see [1, 2] or [10].

A brief description of each of the probe summarization options available in **ArrayAssist** is given below. Some of these algorithms are native implementations within **ArrayAssist** and some are directly based on the affymetrix codebase. The exact details are described in the table below.

RMA	Implemented in **ArrayAssist**	Validated against R
GCRMA	Implemented in **ArrayAssist**	Validated against default GCRMA in R
MAS5	Licensed from Affymetrix	Validated against Affymetrix Data
PLIER	Summarization licensed from Affymetrix, Normalization implemented in **ArrayAssist**	Validated against Affymetrix Data
LiWong	Implemented in **ArrayAssist**	Validated against R
Absolute Calls	Licensed from Affymetrix	Validated against Affymetrix Data

Masked Probes and Outliers. Finally, note that CEL files have masking and outlier information about certain probes. These masked probes and outliers are removed.

The RMA (Robust Multichip Averaging) Algorithm

The RMA method was introduced by Irazarry et al. [1, 2] and is used as part of the RMA package in the Bioconductor suite. In contrast to MAS5, this is a PM based method. It has the following components.

Background Correction. The RMA background correction method is based on the distribution of MM values amongst probes on an Affymetrix array. The key observation is that the smoothened histogram of the $\log(MM)$ values exhibits a sharp normal-like distribution to the left of the mode (i.e., the peak value) but stretches out much more to the right, suggesting that the MM values are a mixture of non-specific binding and background noise on one hand and specific binding on the other hand. The above peak value is a natural estimate of the average background noise and this can be subtracted from all PM values to get background corrected PM values. However, this causes the problem of negative values. Irizarry et al. [1, 2] solve the problem of negative values by imposing a positive distribution on the background corrected values. They assume that each observed PM value O is a sum of two components, a signal S which is assumed to be exponentially distributed (and is therefore always positive) and a noise component N which is normally distributed. The background corrected value is obtained by determining the expectation of S conditioned on O which can be computed using a closed form formula. However, this requires estimating the decay

parameter of the exponential distribution and the mean and variance of the normal distribution from the data at hand. These are currently estimated in a somewhat ad-hoc manner.

Normalization. The RMA method uses Quantile normalization. Each array contains a certain distribution of expression values and this method aims at making the distributions across various arrays not just similar but identical! This is done as follows. Imagine that the expression values from various arrays have been loaded into a dataset with probesets along rows and arrays along columns. First, each column is sorted in increasing order. Next, the value in each row is replaced with the average of the values in this row. Finally, the columns are unsorted (i.e., the effect of the sorting step is reversed so that the items in a column go back to wherever they came from). Statistically, this method seems to obtain very sharp normalizations [3]. Further, implementations of this method run very fast.

Probe Summarization. RMA models the observed probe behavior (i.e., $\log(PM)$ after background correction) on the log scale as the sum of a probe specific term, the actual expression value on the log scale, and an independent identically distributed noise term. It then estimates the actual expression value from this model using a robust procedure called *Median Polish*, a classic method due to Tukey.

The GCRMA Algorithm

This algorithm was introduced by Wu et al [7] and differs from RMA only in the background correction step. The goal behind its design was to reduce the bias caused by not subtracting MM in the RMA algorithm. The GCRMA algorithm uses a rather technical procedure to reduce this bias and is based on the fact that the non-specific affinity of a probe is related to its base sequence. The algorithm computes a background value to be subtracted from each probe using its base sequence. This requires access to the base sequences. **ArrayAssist** packages all the required sequence information into the Chip Information Package, so no extra file input is necessary.

The Li-Wong Algorithm

There are two versions of the Li-Wong algorithm [6], one which is $PM-MM$ based and the other which is PM based. Both are available in the dChip software. **ArrayAssist**has only the $PM - MM$ version.

Background Correction. No special background correction is used by the

ArrayAssist implementation of this method. Some background correction is implicit in the $PM - MM$ measure.

Normalization. While no specific normalization method is part of the Li-Wong algorithm as such, dChip uses *Invariant Set* normalization. An *invariant set* is a a collection of probes with the most conserved ranks of expression values across all arrays. These are identified and then used very much as spike-in probesets would be used for normalization across arrays. In **ArrayAssist**, the current implementation uses Quantile Normalization [3] instead, as in RMA.

Probe Summarization. The Li and Wong [6] model is similar to the RMA model but on a linear scale. Observed probe behavior (i.e., $PM - MM$ values) is modeled on the linear scale as a *product* of a probe affinity term and an actual expression term along with an additive normally distributed independent error term. The maximum likelihood estimate of the actual expression level is then determined using an estimation procedure which has rules for outlier removal. The outlier removal happens at multiple levels. At the first level, outlier arrays are determined and removed. At the second level, a probe is removed from all the arrays. At the third level, the expression value for a particular probe on a particular array is rejected. These three levels are performed in various iterative cycles until convergence is achieved. Finally, note that since $PM - MM$ values could be negative and since **ArrayAssist** outputs values always on the logarithmic scale, negative values are thresholded to 1 before output.

The Average Difference and Tukey-BiWeight Algorithms

These algorithms are similar to the MAS4 and MAS5 methods [4] used in the Affymetrix software, respectively.

Background Correction. These algorithm divide the entire array into 16 rectangular zones and the second percentile of the probe values in each zone (both PM's and MM's combined) is chosen as the background value for that region. For each probe, the intention now is to reduce the expression level measured for this probe by an amount equal to the background level computed for the zone containing this probe. However, this could result in discontinuities at zone boundaries. To make these transitions smooth, what is actually subtracted from each probe is a weighted combination of the background levels computed above for all the zones. Negative values are avoided by thresholding.

Probe Summarization. The one-step Tukey Biweight algorithm combines together the background corrected $\log(PM - MM)$ values for probes within a probe set (actually, a slight variant of MM is used to ensure that $PM - MM$ does not become negative). This method involves finding the median and weighting the items based on their distance from the median so that items further away from the median are down-weighted prior to averaging.

The Average Difference algorithm works on the background corrected $PM - MM$ values for a probe. It ignores probes with $PM - MM$ intensities in the extreme 10 percentiles. It then computes the mean and standard deviation of the $PM - MM$ for the remaining probes. Average of $PM - MM$ intensities within 2 standard deviations from the computed mean is thresholded to 1 and converted to the log scale. This value is then output for the probeset.

Normalization. This step is done after probe summarization and is just a simple scaling to equalize means or trimmed means (means calculated after removing very low and very high intensities for robustness).

The PLIER Algorithm

This algorithm was introduced by Hubbell [5] and introduces a integrated and mathematically elegant paradigm for background correction and probe summarization. The normalization performed is the same as in RMA, i.e., Quantile Normalization. After normalization, the PLIER procedure runs an optimization procedure which determines the best set of weights on the PM and MM for each probe pair. The goal is to weight the PMs and MMs differentially so that the weighted difference between PM and MM is non-negative. Optimization is required to make sure that the weights are as close to 1 as possible. In the process of determining these weights, the method also computes the final summarized value.

Comparative Performance

For comparative performances of the above mentioned algorithm, see [1, 2] where it is reported that the RMA algorithm outperforms the others on the GeneLogic spike-in study [19]. Alternatively, see [10] where all algorithms are evaluated against a variety of performance criteria.

5.5.2 Computing Absolute Calls

ArrayAssist uses code licenced from Affymetrix to compute calls. The Present, Absent and Marginal Absolute calls are computed using a Wilcoxon Signed Rank test on the (PM-MM)/(PM+MM) values for probes within a probeset. This algorithm uses the following parameters for making these calls:

- The *Threshold Discrimination Score* is used in the Wilcoxon Signed Rank test performed on (PM-MM)/(PM+MM) values to determine signs. A higher threshold would decrease the number of false positives but would increase the number of false negatives.
- The second and third parameters are the *Lower Critical p-value* and the *Higher Critical p-value* for making the calls. Genes with p-value in between these two values will be called Marginal, genes with p-value above the Higher Critical p-value will be called Absent and all other genes will be called Present.

Parameters for Summarization Algorithms and Calls

The algorithms MAS5 and PLIER and the Absolute Call generation procedure use parameters which can be seen at *File* $\longrightarrow$*Config*. However, modifications of these parameters are not currently available in **ArrayAssist**. These should be available in the future versions.

5.5.3 GO Computation

Suppose we have selected a subset of significant genes from a larger set and we want to classify these genes according to their ontological category. The aim is to see which ontological categories are important with respect to the significant genes. Are these the categories with the maximum number of significant genes, or are these the categories with maximum enrichment? Formally stated, consider a particular GO term G. Suppose we start with an array of n genes, m of which have this GO term G. We then identify x of the n genes as being significant, via a T-Test, for instance. Suppose y of these x genes have GO term G. The question now is whether there is enrichment for G, i.e., is y/x significantly larger than m/n. How do we measure this significance?

ArrayAssist computes a p-value to quantify the above significance. This p-value is the probability that a random subset of x genes drawn from

the total set of n genes will have y or more genes containing the GO term G. This probability is described by a standard hypergeometric distribution (given n balls, m white, n-m black, choose x balls at random, what is the probability of getting y or more white balls). **ArrayAssist** uses the hypergeometric formula from first principles to compute this probability.

Finally, one interprets the p-value as follows. A small p-value means that a random subset is unlikely to match the actually observed incidence rate y/x of GO term G, amongst the x significant genes. Consequently, a low p-value implies that G is enriched (relative to a random subset of x genes) in the set of x significant genes.

NOTE: The same gene may be counted repeatedly in GO p-value computation due to association with multiple probesets. Currently, the computations don't take this factor into account.

Chapter 6

Importing EXON Data

6.1 Analyzing Affymetrix Exon Chips

ArrayAssist has workflows specifically crafted for analyzing the all exon chips from affymetrix. This section contains two major subsections.

- Section Importing and Analyzing Exon Data, a description of the exon data import and analysis process.
- Section Example Tutorial on Exon Analysis, an example tutorial to get first-time users acquainted with the exon workflow.

6.1.1 Space Requirements

Please note the following special requirements for working with exon CEL files which contain much larger amounts of data than the largest Affymetrix 3'IVT chips.

Disk Space Requirement. Please make sure that the amount of disk space available is at least 200MB per CEL file you wish to process. This space must be available on the disk drive in which your project is being saved. Probset summarization will stop midway if this amount of space is not available.

Memory Setup. It is recommended that you have a 2GB RAM machine for processing Exon files. It is also recommended that you make the following modification in the installation-folder/bin/packages/properties.txt file which can be edited using Wordpad or any other text editor: in the java.options line, modify -Xmx1024m to -Xmx1500m. Shut down **ArrayAssist** before making this change and relaunch after the change is made for

the change to take effect. This change allows Java to use a larger amount of memory on your machine.

Note that on some machines, launching **ArrayAssist** after making this change will cause all text to blank out; in such cases, you will need to set your hardware acceleration configuration on your machine (on Windows XP, go to *My Computer* $\longrightarrow$*Display* $\longrightarrow$*Settings* $\longrightarrow$*Advanced* $\longrightarrow$*Troubleshoot* and set the acceleration to the third bar from the left).

In addition, on some rare machines, **ArrayAssist** will not start up at all with the above change. The reason for this is the presence of some other applications having reserved certain memory slots. In such a situation, the best course of action would be to reduce the -Xmx value above to a lower value. You will need to identify the highest value for which **ArrayAssist** starts up via trial and error. This will affect the number of CEL files that can be processed in one project. Alternatively, use a fresh new machine without other applications installed.

Memory Requirement. ArrayAssist has been optimized to perform probeset summarization and generate signal values for all 1.4 million probesets on any number of arrays irrespective of the amount of RAM available. However, memory limits kick in for viewing and analyzing these signal values.

On Windows XP, generating probeset signal values for all probesets can be done for up to 150 arrays, leaving about 600MB for further analysis. The rest of the memory usage depends upon how much filtering happens at each stage. Assuming DABG and Significance Analysis filters reduce the number of probesets of interest to about 300,000 (i.e., the total number of probesets over all transcripts which contain at least one significant probeset), Transcript Summarization will run and leave another 200MB or so of space. At this point the project can be saved and the probeset summarized data deleted leaving plenty of space for all further analysis. The full standard exon workflow on **ArrayAssist** has indeed been tested on up to a 150 arrays with the *All Probe Sets* option and the entire workflow run on a 2GB RAM machine with the -Xmx value set to 1550m.

Note also that if only probeset signals need to be generated and viewed and no further analysis needs to be performed then the number of CEL files can go to above 200. Finally, note that on Fedora Core 3 Linux machines with more than 2GB or RAM, the -Xmx setting can be made larger and therefore a larger number of CEL files can be supported.

Keeping Track of Memory Usage. Finally, keep a watch on the memory monitor at the bottom right of **ArrayAssist** , which shows a message stat-

ing that the application is using x MB of y. Click on the garbage can icon at the bottom right occassionally to force **ArrayAssist** to release memory. If y starts getting close to the limit specified in -Xmx option above then make sure you save your project and delete the main probeset summarized dataset, keeping only the splicing analysis dataset and all children datasets thereof. This will provide plenty of memory for further downstream operations. An operation that demands a large amount of memory causing application memory to cross the -Xmx limit set above could cause an application crash.

6.2 Importing and Analyzing Exon Data

Use the following command to import CEL files into **ArrayAssist** to create a new Exon project.

File⟶New Affymetrix Exon Project

NOTE: Affymetrix CEL and CHP files are available in two formats, the Affymetrix GeneChip Command Console compliant data file (AGCC) files; and Extreme Data Access compliant data (GCOS
XDA) files. **ArrayAssist** 5.1 uses the recently released Affymetrix Fusion SDKs that supports both AGCC and XDA format CEL and CHP files. However the older Affymetrix GDAC SDKs are also avaliable in **ArrayAssist**. By default,

6.2.1 Selecting CEL/CHP Files

The first step in creating the project is to provide a project name and folder path and then select CEL files of interest. The project folder will be used to save the .avp project file in addition to several pieces of intermediate information created while processing CEL files.

To select files, click on the *Choose File(s)* button, navigate to the appropriate folder and select the files of interest. Use Left-Click to select the first file, Ctrl-Left-Click to select subsequent files, and Shift-Left-Click for a contiguous set of files. Once the files are selected, click on *OK*. If you wish to select files from multiple directories or multiple contiguous chunks of files from the same directory, you can repeat the above exercise multiple times, each time adding one chunk of files to the selection window. You can remove already chosen files by first selecting them (using Left-Click , Ctrl-Left-Click

and Shift-Left-Click , as above) and then clicking on the *Remove Files* button. After you have chosen the right files, hit the *Next* button. Note that the dataset will be created with each column corresponding to one CEL file or one experiment. The order of the columns in the dataset will be the same as the order in which they occur in the selection interface If you want the columns in the dataset to be in any specific order, you should order them here appropriately.

NOTE: The space required per Human Exon CEL file is approximately 200MB. If the required amount of space in not available, CEL file processing could abort midway.

6.2.2 Getting Chip Information Packages

To import Exon CEL files, you will need the *Chip Information Package* for your chip of interest. This package contains probe layout information derived from the CDF file as well as gene annotation information derived from the NetAffx comma separated annotation file. You can fetch this file using *Tools⟶Update Data Library*.

NOTE: **Chip Information Packages** could change every quarter as new gene annotations are released on NetAffx by Affymetrix. These will be put up on the **ArrayAssist** update server. **ArrayAssist** will directly keep track of the latest version available on **ArrayAssist** update server. When **ArrayAssist** launches, it will check the version available on the local machine with the version on the server. If a newer version has been deployed on the server, then, on starting, **ArrayAssist** will launch the update utility with the specific libraries check and marked for update.
Each project stores the generation date of the *Chip Information Package*. If newer libraries are available on the tool, when the project is opened, you will be prompted with a dialog asking you whether you want to refresh the annotations. Clicking on *OK* will update all the annotations columns in the project. You can also refresh the annotations after the project is loaded from the *Refresh Annotations* link in the workflow.

6.3 Running the Affymetrix Exon Workflow

When the new Exon project is created after proceeding through the above *File⟶New Affymetrix Exon Project* wizard, **ArrayAssist** with open a new

project with the following view:

The Data Description View: This view shows a list of CEL files imported in the panel on the left. The *File Header* tab shows the file header containing some statistics for the file selected on the left panel.

You are now ready to run the Affymetrix Exon Workflow. The Affymetrix Exon Workflow Browser contains all typical steps used in the analysis of Affymetrix microarray data. These steps will output various datasets and views. The following note will be useful in exploring these views.

NOTE: Most datasets and views in **ArrayAssist** are *lassoed*, i.e., selecting one or more rows/columns/points will highlight the corresponding rows/columns/points in all other datasets and views. In addition, if you select probesets from any dataset or view, signal values and gene annotations for the selected probesets can be viewed using *View* $\longrightarrow$*Lasso* (you may need to customize the columns visible on the Lasso view using Right-Click *Properties*).

6.3.1 Providing Experiment Grouping Information

Experiment Factors and Groups. Click on the *Experiment Grouping* link in the workflow browser. The *Experiment Grouping* view which comes up will initially just have the CEL/CHP file names. The task of grouping will involve providing more columns to this view containing *Experiment Factor* and *Experiment Grouping* information. A Control vs. Treatment type experiment will have a single factor comprising 2 groups, Control and Treatment. A more complicated Two-Way experiment could feature two experiment factors, genotype and dosage, with genotype having transgenic and non-transgenic groups, and dosage having 5, 10, and 50mg groups. Adding, removing and editing experiment factors and associated groups can be performed using the icons described below.

Reading Factor and Grouping Information from Files. Click on the Read Experiment Grouping from File  icon icon to read in all the Experiment Factor and Grouping information from a tab or comma separated text file. The file should contain a column containing CEL/CHP file names; in addition, it should have one column per factor containing the grouping information for that factor. Here is an example tab separated file. The result of reading this tab file in is the new columns corresponding to each factor in the Experiment Grouping view.

```
#comments
#comments
filename genotype dosage
A1.CEL    NT       0
A2.CEL     T       0
A3.CEL    NT      20
A4.CEL     T      20
A5.CEL    NT      50
A6.CEL     T      50
```

Adding a New Experiment Factor. Click on the Add Experiment Factor  icon to create a new experiment factor and give it a name when prompted. This will show the following view asking for grouping information corresponding to the experiment factor at hand. The CEL/CHP files shown in this view need to be grouped into groups comprising biological replicate arrays. To do this grouping, select a set of CEL/CHP files, then click on the *Group* button, and provide a name for the group. Selecting CEL/CHP files use Left-Click , Ctrl-Left-Click , and Shift-Left-Click , as before.

Editing an Experiment Factor. Select the experiment factor you want to edit, by clicking on the respective factor column. This column will be selected. Click on the *Edit Experiment Factor*  icon to edit an Experiment Factor. This will pull up the same grouping interface described in the previous paragraph. The groups already set here can be changed on this page.

Remove an Experiment Factor. Click on the Remove Experiment Factor  icon to remove an Experiment Factor.

6.3.2 Running Probe Summarization Algorithms

Currently **ArrayAssist** supports two main algorithms, the ExonRMA algorithm and the ExonPLIER algorithm. For more technical details of these algorithms, see Section Algorithm Technical Details below.

These algorithms can either be run on *All* probesets or on specific subsets of probesets (which are labelled *Core*, *Extended* and *Full*, respectively). The extended option includes *Core* and *Extended* probesets and the *Full* option includes *Core*, *Extended* and *Full* probesets. The *All* option will output 1.4 million probesets, the *Full* option also outputs about 1,400,000 probesets, the *Extended* option outputs about 800,000, and the *Core* option outputs

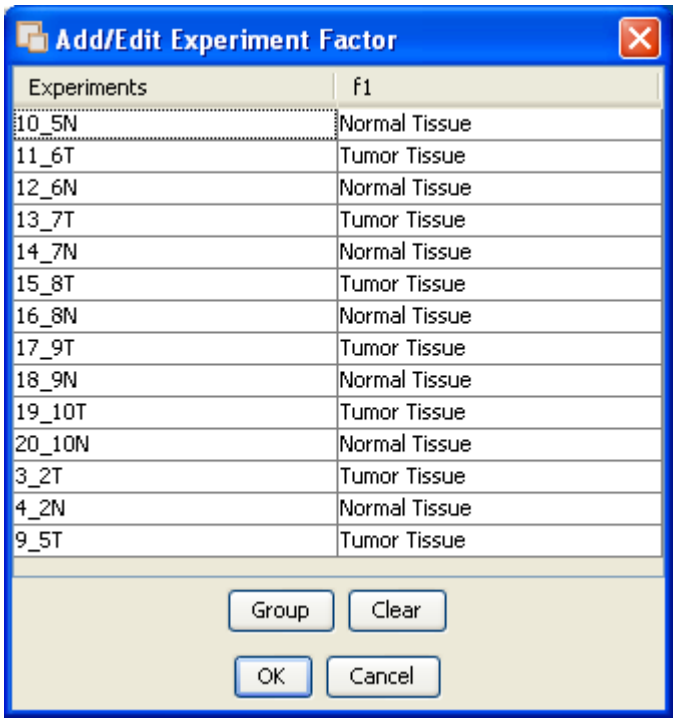


Figure 6.1: Specify Groups within an Experiment Factor

about 300,000. The default is set to *Extended*. The *All* option is redundent since it is the same as the *Full*, however this option has been retained.

In addition, both algorithms allow for a choice of background probes; users can choice either only antigenomic background probes or genomic background probes or both. The default is set to Antigenomic. The PM-GCBG option will perform background correction using these background probes and the PM option will not use these background probes at all.

A variance stabilization addition of 16 is done to both algorithms; this amount can be specified on the summarization dialog.

Both algorithms give you the choice to perform quantile normalization. The default is to perform quantile normalization. If you do not want to perform quantile normalization, uncheck this option.

The result of this step is a new *Summarized Probeset* dataset containing probeset signal values on the *log* scale (in contrast to the Affymetrix Expression workflow in **ArrayAssist** which used the linear scale).

Quality Assessment

One you have a *Summarized* dataset, the next step would be to check for sample and data quality. **ArrayAssist** provides the following workflow steps to do this.

NOTE: Remember to select a Probeset Summarized dataset on the navigator before running one of the following steps.

Hybridization Quality Assessment Plots Clicking on this link will output two types of sample and hybridization quality views:

The *Poly-A Controls* view is used to monitor the entire target labeling process. Lys, phe, thr, and trp are B. subtilis genes that have been modified by the addition of poly-A tails and then cloned into pBluescript vectors which contain T3 promoter sequences. Amplifying these poly-A controls with T3 RNA polymerase will yield sense RNAs, which can be spiked into a complex RNA sample, carried through the sample preparation process, and evaluated like internal control genes. There is one profile for each array, with the Legend at the bottom-right showing which profile corresponds to which array.

The *Hybridization Controls* view depicts the hybridization quality. Hybridization controls are composed of a mixture of biotin-labelled cRNA transcripts of bioB, bioC, bioD, and cre prepared in staggered concentrations

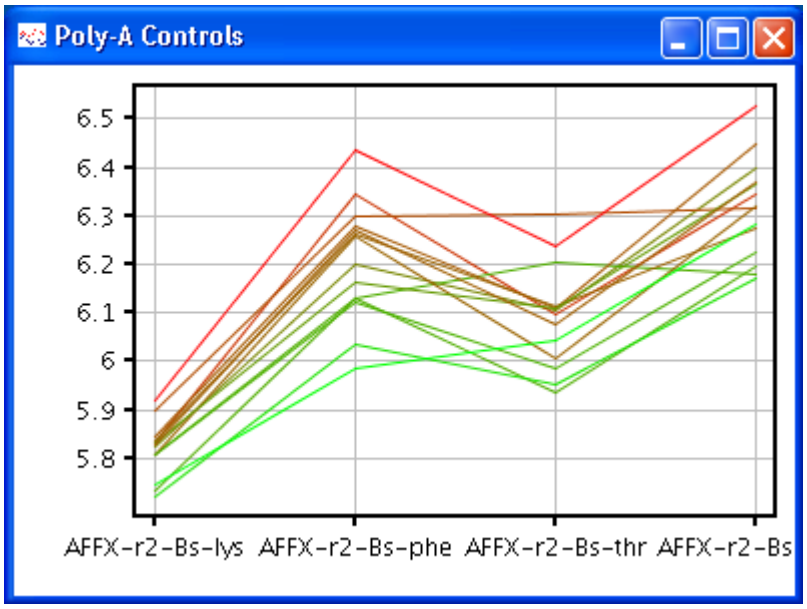


Figure 6.2: Poly-A Control Profiles

(1.5, 5, 25, and 100pm respectively). This mixture is spiked-in into the hybridization cocktail. bioB, bioC, bioD and cre must appear in increasing concentrations. The Hybridization Controls view shows the signal value profiles of these transcripts (only 3' probesets are taken). There is one profile for each array, with the Legend at the bottom-right showing which profile corresponds to which array.

Principal Component Analysis on Arrays. This link will perform principal component analysis on the arrays. It will show the standard PCA plots (see PCA for more details). The most relevant of these plots used to check data quality is the *PCA scores* plot, which shows one point per array and is colored by the Experiment Factors provided earlier in the *Experiment Grouping* view. This allows viewing of separations between groups of replicates. Ideally, replicates within a group should cluster together and separately from arrays in other groups. The *PCA scores* plot can be color customized via Right-Click *Properties*. All the Experiment Factors should occur here, along with the Principal Components E0, E1 etc. The PCA Scores view is *lassoed*, i.e., selecting one or more points on this plot will highlight the corresponding columns (i.e., arrays) in all the datasets and views. Further details on running PCA appear in the chapter on PCA.

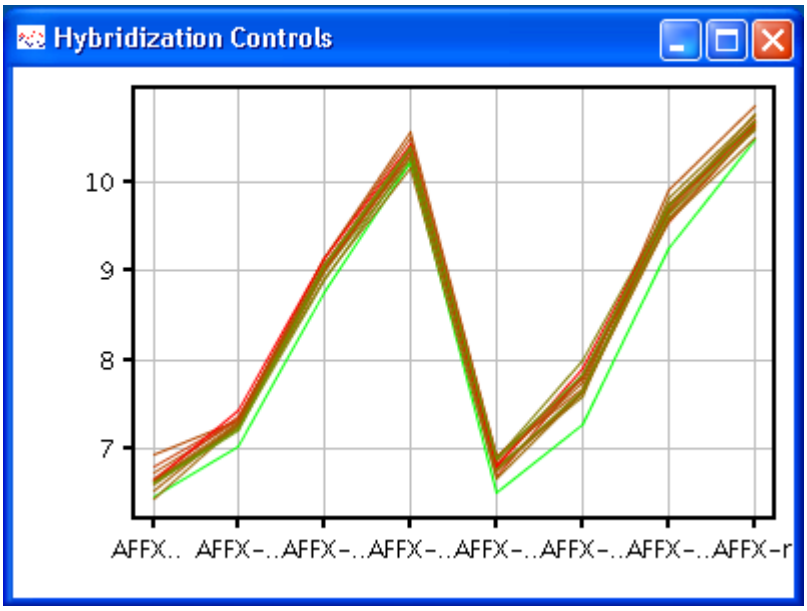


Figure 6.3: Hybridization Control Profiles

Correlation Plots. This link will perform correlation analysis across arrays. The correlation coefficient for a pair of arrays is defined as

$$\sum_i [(a_i - \mu_a) * (b_i - \mu_b)]/n(\sigma_a * \sigma_b)$$

where a_i are the signals in array a, b_i are the signals in array b, μ and σ are the respective means and standard deviations, and n is the number of items in each array.

This step finds the correlation coefficient for each pair of arrays and then displays these in two forms, one in textual form as a correlation table view, and other in visual form as a heatmap. The labels in the heat map can be colored by the experimental group of the array name via Right-Click *Properties*. The intensity levels in the heatmap can also be customized here. The table view itself can be exported via Right-Click *Export as Text*. Note that unlike most views in **ArrayAssist**, the correlation views are not *lassoed*, i.e., selecting one or more rows/columns here will not highlight the corresponding rows/columns in all the other datasets and views.

Sometimes it is useful to reorder the arrays before performing this analysis so that the heat map patterns are more discernible. Additionally, you

may want to cluster the arrays based on correlation. To do this, export the correlation text view as text, then open it via *File*⟶*Open*, and then use *Cluster*⟶*Hier* to cluster. Row labels on the resulting dendrogram can then be colored based on Experiment Factors using Right-Click *Properties*.

Summary Statistics. This link will show summary statistics for each array which includes the mean, the median, the percentiles, the trimmed mean and the number of outliers in each array.

6.3.3 DABG Filtering

Once data is summarized, probesets below noise level can be filtered out using the DABG (Detection above Background) filter. This will run the DABG (detection above background) method from the Affymetrix Exact 1.1 software. This method returns a p-value for each probeset on each array, with low p-values indicating signal significance.

ArrayAssist does not explicitly output the p-value to save space; instead, **ArrayAssist** asks for a filter criterion and creates a new filtered probeset dataset containing only probesets which satisfy the filter condition. The filter condition requires at least a certain number of arrays to have a low p-value for that probeset.

If you want to see the DABG p-values explicitly, use the DABG link in the Utilities section of the Affymetrix Exon Workflow Browser.

6.3.4 Probeset Statistical Significance Analysis

This section allows you to filter probesets using a battery of statistical tests including T-Tests, Mann-Whitney Tests, Multi-Way ANOVAs and One-Way Repeated Measures tests. The purpose of this section is to identify transcripts which have at least one probeset which is expressed differentially across experimental groups.

Clicking on the *Significance Analysis Wizard* will launch the full wizard which will guide you through the various testing choices for testing each probeset for significance. Details of these choices appear in The Differential Expression Analysis Wizard, along with detailed usage descriptions. Results of Significance Analysis are presented in views and datasets described below. All of these appear under the *Diffex* node in the navigator as shown below.

The Statistics Output Dataset. This dataset contains the p-values and fold-changes for each probeset (and other auxiliary information), generated by Significance Analysis.

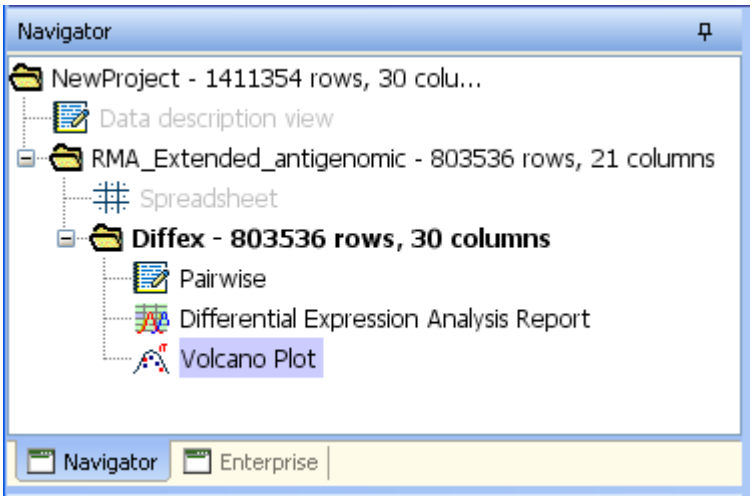


Figure 6.4: Navigator Snapshot Showing Significance Analysis Views

The Differential Expression Analysis Report. This report shows the test type and the method used for multiple testing correction if any and the corresponding p-values. In addition, it shows the distribution of genes across p-values and fold-changes in a tabular form. For T-Tests, each table cell shows the number of genes which satisfy the corresponding p-value and fold-change cutoffs. For ANOVAs, each table cell shows the number of genes which satisfy the corresponding p-value cutoff only. For multiple T-Tests, the report view will present a drop down box which can be used to pick the appropriate T-Test. Clicking on a cell in these tables will select and lasso the corresponding genes in all the views. Finally, note that the last row in the table shows some *Expected by Chance* numbers. These are the number of genes expected by pure chance at each p-value cut-off. The aim of this feature is to aid in setting the right p-value cutoff. This cut-off should be chosen so that the number of gene expected by chance is much lower then the actual number of genes found (see The Differential Expression Analysis Wizard for details).

The Volcano Plot. This plot shows a scatter plot the log of p-value against the log of fold-change. Probesets with large fold-change and low p-value are easily identifiable on this view. The properties of this view can be customized using Right-Click *Properties*.

Filtering on p-values and Fold Changes. There are four ways to filter.

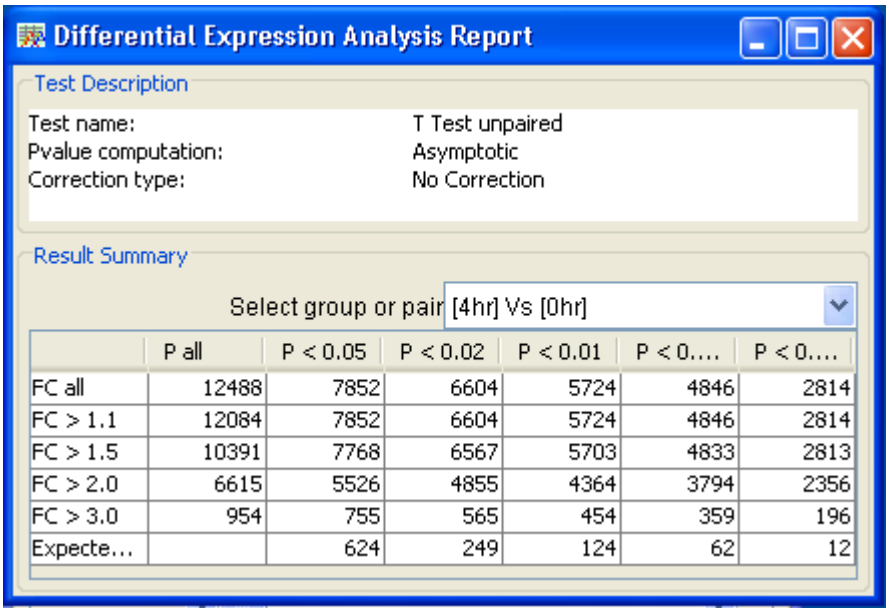


Figure 6.5: Differential Analysis Report

The first and simplest option uses the *Transcripts with Significant Probesets* link in the workflow browser. Fill in cut-offs for p-value, fold-change and regulation (up, down or both). Conditions on the various groups shown in this dialog are combined via an "and", i.e., all of the specified cut-offs must be satisfied. A new dataset will be created with the relevant probesets. In addition, further probesets will be included to make this dataset transcript-complete, i.e., all probesets for a transcript will be included if any one of the probesets passes the filter.

The second way is to click on a relevant cell of the Differential Expression Analysis Report view. This will select all corresponding probesets in all open views. You can then use the *Data* $\longrightarrow$*Create Subset* $\longrightarrow$*Create Subset from Selection* operation to create a new subset dataset from this link.

The third way is to go to the statistics output dataset, sort the p-value or fold-change columns, select as many rows from this table as necessary, and again create a new dataset from the selection.

The fourth and most powerful way is useful in complex scenarios. Consider situations where you do two separate statistical tests and want to identify genes with a p-value less than say 0.05 in one experiment and p-value greater than .1 in the other. Use the *Data* $\longrightarrow$*Columns* $\longrightarrow$*New Column*

Using Formula to create a new column in the Statistics Output Dataset containing values 1 (relevant) and 0 (not relevant). Then sort this column so the 1's come to the top, select all the rows with 1s and create a new dataset from selection. To see examples of formulae and tips on usage of the New Column with Formula command, see Section on Create New Column using Formula.

Note that this subset dataset created by the *Create Subset from Selection* command will not be transcript-complete, i.e., it could have some but not all probesets for any particular transcript. Downstream splicing analysis may require transcript-completeness so one can compare and contrast all probesets for a particular transcript. The downstream Transcript summarization step will automatically perform an expansion on transcripts (i.e., consider all probesets for each relevant transcript). Alternatively, you can use the *Expand on Transcript* link in the Utilities section of the workflow to create a new dataset which is transcript-complete.

6.3.5 Gene Level Analysis

This section of the Exon workflow provides for generating transcript signal values and running statistical tests on transcript signals and splicing indices (defined as the difference between the probeset and the transcript log scale signal).

Gene Level Summarization.

This link will perform transcript summarization on the current dataset containing a subset of probesets resulting from the previous workflow steps. Summarization will be performed for each transcript represented in this dataset; all probesets in each of these transcripts (and not just probesets present in the current dataset) will be used for summarization. Probesets without a transcript label will be dropped.

The transcript summarization process will automatically choose the same algorithm (i.e., exonRMA or exonPLIER) and associated parameters as those used for probeset summarization earlier.

The resulting dataset created (called the Splicing Analysis Dataset) will have a row for each of the probesets in each of the relevant transcripts. In addition, it will contain probeset signal columns and the newly obtained transcript signal columns. Finally, it will also contain four chromosome information columns required for further splicing analysis (the chromosome number, start, stop, and strand columns). The dataset that is created will have one row for each probeset and the transcript summarized signal values will be repeated for each of the probesets.

Splicing Indices (defined as the log-scale difference between probeset and the transcript signal) are not automatically computed at this step to save space. All subsequent links which work on splicing indices will compute these indices on demand. A separate link is provided in the Utilities section for explicit computation of splicing indices.

Note that once you have the Splicing Analysis Dataset, you can save the project and delete the Probeset Summarized dataset to free space for further analysis.

Baseline Transforming Gene Level Data

Baseline transformation of any data table in ArrayAssist can be done using the `exon_baseline_transform.py` script found in the `<INSTALL_DIR>/samples/scripts` folder.

To baseline transform a transcript summarized data table in an ArrayAssist Exon project select the desired data table in the navigator. From the drop-down menu select Tools ⟶Script Editor. Use first button to *Open a script file*, browse to the `<INSTALL_DIR>/samples/scripts/exon_baseline_transform.py` file and press Open. Click on the Run  icon button on the Script Edition tool bar. This will invoke the script dialog.

In the script dialog select the Columns for Computing Baseline Mean. Columns selected will be averaged. Columns for Applying Baseline Transform allows users to choose which columns of data will be baseline transformed. If using transcript summarized data (which is in log2 space) ensure that the Option for Baseline Transform is set to **Subtract Baseline Average**. Applying these settings will result in a child **Baseline Transformed** dataset in the navigator.

Gene Level Significance Analysis This step performs statistical testing on transcripts. The usage is very similar to that of the probeset significance analysis section earlier (section Probeset Statistical Significance Analysis); the main difference is that this step runs on transcript signal values rather than probeset signal values. The significance analysis report, the volcano plot, and the statistics dataset will contain transcripts rather than probesets.

Note that selecting transcripts on one of these views will not select all probesets for all selected transcripts in the other views which represent probsets; rather, only the first probeset in each transcript gets selected for technical reasons. There are two ways to select all probesets corresponding to selected transcripts here. The first way is to save a genelist using the *Create*

Probeset List from Selection link in the workflow browser; choose Transcript Cluster Id as the genelist *Mark*. Then go to any probeset level dataset in the navigator and double click on the genelist; all probesets corresponding to transcript ids saved in the selected genelist will get selected. The second way is to use the *Expand on Transcripts* link in the utilities section of the workflow browser to create a new dataset with probesets for the selected transcripts.

The *Identify Significant Transcripts* link allows the user to choose p-value and fold-change cut-offs and creates a new dataset which automatically contains all probesets for all selected transcripts. In addition, as mentioned in the corresponding filtering step description in Section Probeset Statistical Significance Analysis, there are other methods to filter as well; these involve selecting relevant transcripts from the Statistics output dataset or the Differential Expression Analysis report and then creating a new sub-dataset by using the *Expand on Transcript* link in the utilities section.

6.3.6 Splicing Index Analysis

Significance Analysis on Splicing Indices.

This step performs statistical testing on transcripts. The usage is very similar to that of the probeset significance analysis section earlier (section Probeset Statistical Significance Analysis); the main difference is that this step runs on splicing index values (the log scale difference between probeset and transcript signals) rather than probeset signal values. The significance analysis report, the volcano plot, and the statistics dataset will indicate p-values and fold-changes for splicing indices. The filtering steps to identify transcripts with at least one splicing-significant probeset are identical to those in Section Probeset Statistical Significance Analysis.

6.3.7 Views on Splicing Analysis

A set of views for splicing analysis provided in this section is listed below. these views are hepful to visualize the splicing index analysis and identify genes of interest. All these run on the *Splicing Analysis dataset* created by the *Transcript Summarization* link.

Differential Transcript vs Differential Splicing. This view runs on any *Splicing Analysis Dataset* which contains a set of probesets and shows a scatter plot of differential transcript signal vs differential splicing index for each probeset. The differences can be performed between two selected

arrays or between two experimental groups. The probesets in the plot are segregated by chromosome; the chromosome selection panel appears at the bottom. In addition, probesets in a plot are colored by their transcript ids, so probesets belong to the same transcript appear in the same color. The right-click properties on this plot can be used to color-by exon id instead as well.

A filter to view only those transcripts which have low differential transcript value but contain at least one probeset with a high differential splicing value can also be set up in this wizard. Note that differential values are on the log-scale so a value of 1 corresponds to a 2 fold change.

Differential Splicing Index along Chromosome. This view runs on a *Splicing Analysis Dataset*, containing a set of probesets and shows a scatter plot of differential splicing index for each probeset plotted against the probeset chromosome start location. The differential can be performed between two selected arrays or between two experimental groups. The probesets in the plot are segregated by chromosome; the chromosome selection panel appears at the bottom. In addition, probesets in a plot are colored by their exon ids, so probesets belong to the same exon appear in the same color.

A typical usage scenario involves selecting a transcript on the Differential Transcript vs Differential Splicing view and viewing that transcript in this plot. To do this you must move to the relevant chromosome and zoom in on the yellow dots in this plot. You can also set this plot to the *Limit by Selection* option from the right click menu so that only what is selected on the Differential Transcript vs Differential Splicing view is visible in this plot.

Differential Probeset/Transcript Signal along Chromosome. These views is similar to the "differential splicing index along chromosome" view except that they show differential probeset/transcript signal instead.

Profile Plot on Selected Rows. This plot shows either the probeset signal or the splicing index for *selected* probesets in the current dataset across arrays as a profile plot. You will be prompted for the experiment groups you are interested in; you then order the experiment groups and the profile plot comes up in this order.

Heat Map on Selected Rows. This plot shows either the probeset signal or the splicing index for *selected* probesets in the current dataset across arrays as a heat map. You will be prompted for the experiment groups you are interested in; you then order the experiment groups and the profile plot comes up in this order.

6.3.8 Utilities

This section contains various utility functions which are not necessarily required in the primary workflow.

DABG. This will run on in the currently focussed dataset and append the DABG p-values to this dataset; the background probe options (antigenomic/genomic) are chosen automatically from the summarization options which are stored with the dataset. Custom Filters based on these values can be designed using the *Data⟶Column Commands ⟶New Column using a Formula* command to add a new column (see Section 4.1.1). Sorting on this column and selecting the relevant rows of interest will select these probesets in all open views.

Import Annotations. Both Exon and Transcript level annotations available in NetAffx are packaged with the chip information package and can be imported into the currently open dataset via this link. If the dataset contains probesets then probeset annotation is imported. And if the dataset contains transcripts (e.g., the dataset obtained by *Create Compact Transcript Dataset* link in this Utilities section) then transcript level annotation columns is imported.

Create Compact Transcript Dataset. This step runs on a dataset where rows correspond to probesets which contains the probeset and transcript signals, e.g., the *Splicing Analysis Dataset* or any subset thereof. It generates a new dataset where rows correspond to transcripts represented in the input dataset; transcript signal columns are also copied over from the input dataset.

Note that selecting a row in this compact transcript dataset will not automatically select all probesets for this transcript in the other probeset level datasets, rather only the first probeset in the selected transcript is selected for technical reasons. To identify all probesets corresponding to the selected transcripts, use the *Expand on Selected Transcripts* step in this utilities section.

Expand on Selected Transcripts. This step will consider selected transcripts from the current dataset and create a subset of either the main probeset summarized dataset or the *Splicing Analysis Dataset*; this new subset dataset will contain all probesets for the selected transcripts.

Select Genes Based on Keywords. This step asks for a set of columns and a keyword and finds all rows in the current dataset which have a keyword match in the chosen set of columns. All such rows are selected.

6.3.9 Summary of Dataset Types in an Exon Project

There are primarily three types of datasets in an Exon Project.

Probeset Summarized Datasets. These contain one row per probeset, and probeset signals for each probeset. DABG filtering and Probeset Significance analysis can be performed only on such datasets, The transcript summarization link will convert a probeset summarized dataset into a splicing analysis dataset.

Splicing Analysis Datasets. These contain one row per probeset, and probeset as well as transcript signals for each probeset. The first such probeset is created by the Transcript Summarization link. All subsets created thereof also create datasets of this type. Significance Analysis on Transcripts and Splicing Indices as well as the splicing views can be run only on such datasets.

Compact Transcript Datasets. These contain one row per transcript, and transcript signals for each transcript.

6.3.10 Genome Browser

The Genome Browser can be invoked using this link. This browser allows viewing of several static prepackaged tracks. In addition, new tracks can be created based on currently open datasets. For more details on usage, see Section 11.

6.4 Algorithm Technical Details

Here are some technical details of the ExonRMA, PLIER and DABG algorithms.

DABG. All background probes chosen are binned into 25 categories based on their GC count (the number of G,C bases in their corresponding sequences). For each PM probe, its DABG p-value is the fraction of background probes in its corresponding GC bin with a greater signal value; the smaller the p-value the more likely the probe is above background. For each probeset, the p-values of the probes within the probeset are combined into a single p-value as follows.

The p-values of probes within a probeset are converted to logscale, then added up and multiplied by 2 to obtain a test-statistic. Then a chi square probability is computed using this statistic and 2 times the number of probes

in this probeset as the degrees-of-freedom. The resulting value is the DABG value of the probeset.

ExonRMA. ExonRMA does a GC based background correction (described below and performed only with the PM-GCBG option) followed by Quantile normalization followed by a Median Polish probe summarization. The computation takes roughly 30 seconds per CEL file with the *All* option.

The background correction bins background probes into bins based on their GC value and corrects each PM by the median background value in its GC bin. RMA does not have any configurable parameters. The GC based background correction value for a particular PM probe is the median background value its GC bin (see the DABG algorithm above for the definition of GC bins).

ExonPLIER. ExonPLIER does Quantile normalization followed by the PLIER summarization using the PM or the PM-MM options where MM is set to a GC based background estimate described above in ExonRMA; the PM-MM option is used if PM-GCBG is selected. The computation takes roughly 30 minutes per CEL file with the All option. The PLIER implementation and default parameters are those used in the Affymetrix Exact 1.2 package. PLIER parameters can be configured *Tools* $\longrightarrow$*Options* $\longrightarrow$*Affymetrix Algorithms* $\longrightarrow$*ExonPlier*.

6.5 Example Tutorial on Exon Analysis

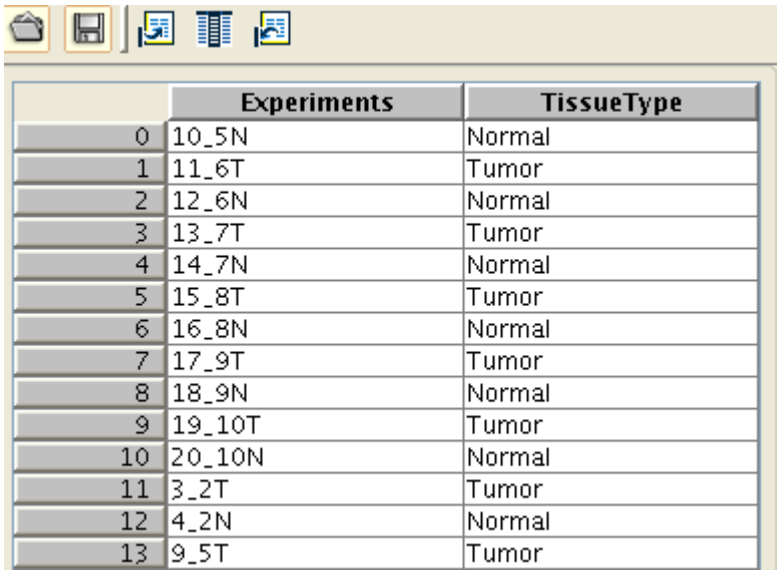
This is an example tutorial which takes you step-by-step through the workflow for analyzing 14 chips run on seven normal samples and seven paired colon cancer tumor samples.

Step 1. Make sure you have at least 1GB of RAM (and preferably 2GB) on your machine.

Step 2. Obtain the exon library pack if you haven't already done so using **Tools**$\longrightarrow$**Update Data Library**, on the resulting screen, click on the *Get Updates* button, then choose the library file which begins with the prefix HuEx-1_0-st.

Step 3. Fetch the 16 CEL files for this tutorial from the colon cancer dataset link

http://www.affymetrix.com/support/technical/sampledata/exon_array_data.affx



	Experiments	TissueType
0	10_5N	Normal
1	11_6T	Tumor
2	12_6N	Normal
3	13_7T	Tumor
4	14_7N	Normal
5	15_8T	Tumor
6	16_8N	Normal
7	17_9T	Tumor
8	18_9N	Normal
9	19_10T	Tumor
10	20_10N	Normal
11	3_2T	Tumor
12	4_2N	Normal
13	9_5T	Tumor

Figure 6.6: Experimental Grouping for the Colon Cancer Dataset

Step 4. Launch **ArrayAssist**. If you have a 2GB RAM machine, you may want to make the memory limit change in the properties.txt file as indicated in the paragraph before launching.

Step 5. Start with the *File* $\longrightarrow$*New Affymetrix Exon Project*. Provide the CEL files of interest and hit next to create a new exon project.

Step 6. Providing experimental grouping is the next step. Clicking on the *Experimental Grouping* link in the exon workflow browser on the right. This will pull up a dialog where the CEL files are listed. The goal now is to provide an experimental group name for each CEL file. Click on the Add Experiment Factor icon to create a new Experiment Factor and give it a name, say "TissueType".

Next, select all CEL files with an _N, then click on the *Group* button, and provide a name for the group, say "Normal". While selecting CEL files uses Left-Click to select a file and Ctrl-Left-Click to add files to the selection. Finally, select all CEL files with an _T, then click on the *Group* button, and provide a name for the group, say "Tumor". Then click OK.

Step 7. Run probeset summarization using the ExonRMA algorithm in the Summarization section of the workflow browser. Use default parameters. This will take about 30 seconds per CEL file on a 3GHz machine. Wait

until the computation finishes and the navigator shows a new *Probeset Summarized Dataset* with about 500,000 rows containing probeset signal values on the log scale.

Step 8. Click on the *Hybridization Quality* link in the Quality Control section of the workflow browser. This should show two plots. The Hybridization controls plot should show an roughly linearly increasing sequence of signal values for the BioB, BioC, BioD and Cre spike-in probesets, as these are spiked-in in doubling concentrations which appear linearly on the log scale.

Step 9. Click on the *PCA* link in the Quality Control section of the workflow browser, then click ok on the resulting dialog. This comes up with two plots, the PCA scores plot and the Eigen Values plot. The PCA scores plot should show one dot for each array colored by the experimental group (see the legend on the bottom left) for details. Change the axes on this plot so you see eigen vectors E0 and E2. This plot shows that the tumors and the normals broadly cluster together and separate from each other, except for 19_10T.

Step 10. Click on the *Correlations Plot* link in the Quality Control section of the workflow browser. In the dialog that comes up, use the up and down buttons on extreme right to reorder the arrays so all tumor arrays come together and all normal arrays come together. Then click OK. This will output 2 views: one contains a spreadsheet with the correlations between each of the arrays. The second contains a graphical color coded view of the same. Right-Click -properties on the graphical view will provide a way to customize the colors and saturation on this graphical view by adjusting the filters. This plot shows that but for 4 arrays, the tumors and normals broadly form homogeneous clumps distinct from each other and the tumors seem more varied than the normals.

Step 11. The next step is to run a DABG (detection above background filter). Click on the *DABG Filter* link on the workflow browser and take the default parameters. This will take some time and create a new filtered dataset in the navigator on the left with all probesets corresponding to transcripts, each of which has at least one probeset detected as being above background (see Section for details).

Step 12. The next step is to run Significance Analysis to identify transcripts which have at least one significant probeset in terms of differential expression. Click on the Probeset Significance Analysis wizard link on the workflow browser. Click the "TissueType" checkbox at the top, and click the "Experiments are Paired" check box at the bottom, and hit Next. On

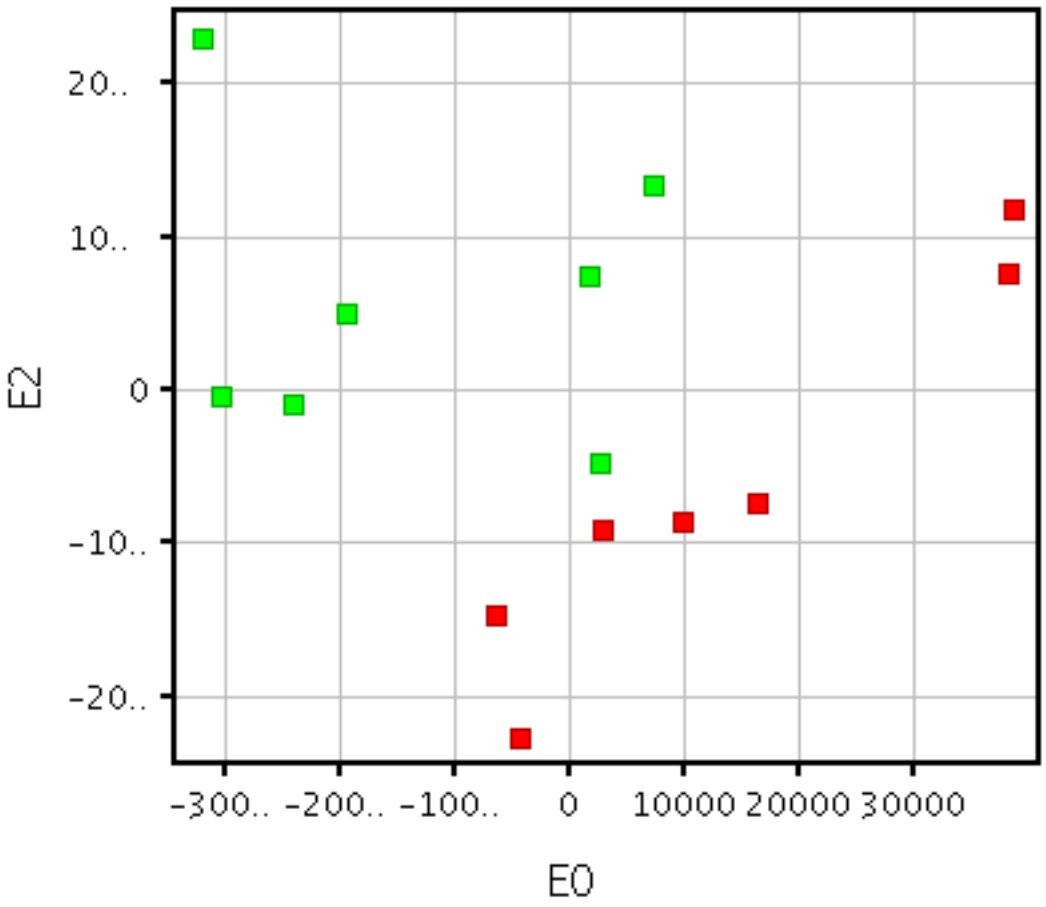


Figure 6.7: PCA Scores Plot of the Colon Cancer Dataset

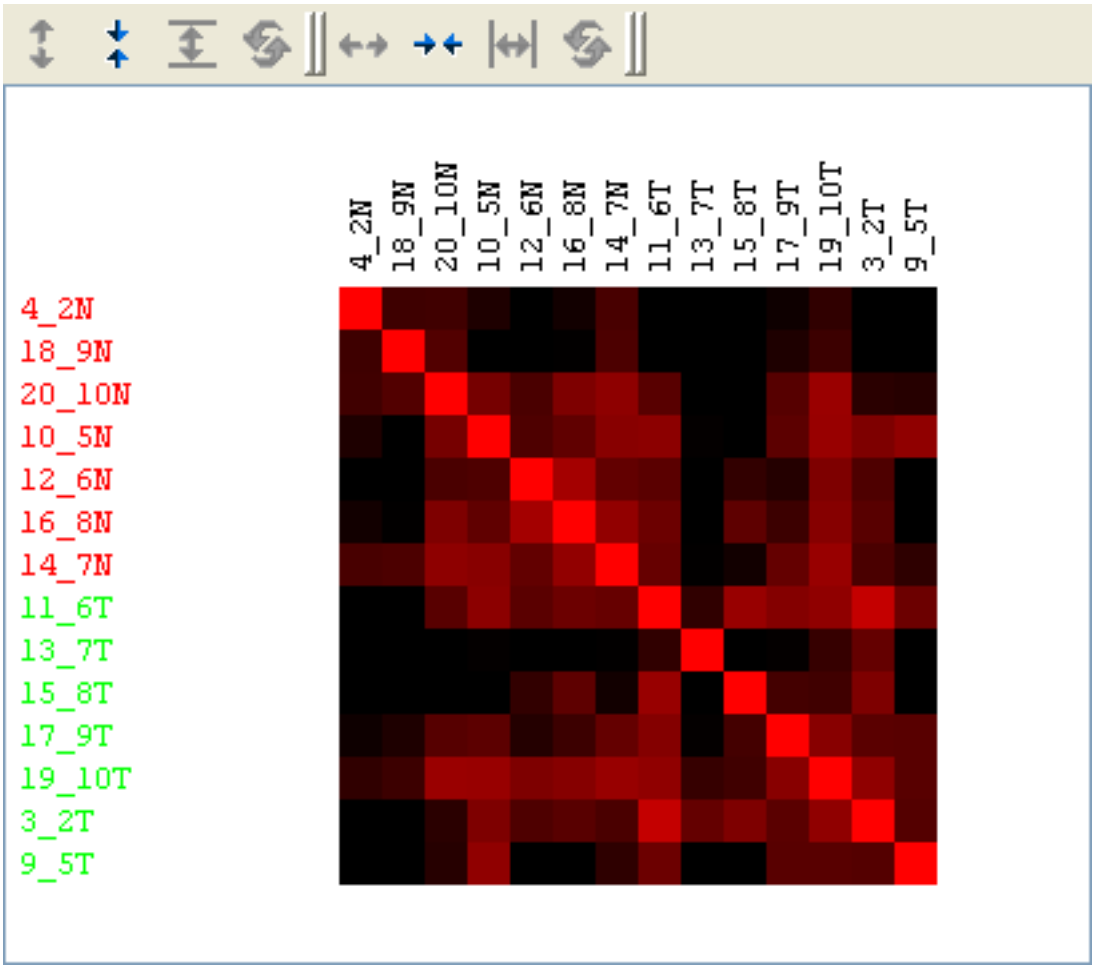


Figure 6.8: Array Correlations on the Colon Cancer Dataset

this next page, provide the pairing between the normals and tumors using the up/down arrows on the right (you need to ensure that _5N and _5T are paired together, as are _6N and _6T etc). Click Next on all subsequent screens leaving default options. This will run a paired T-Test between the normal and tumor groups. Once it finishes running, p-values and fold changes are computed and displayed as a spreadsheet, a volcano plot, as well as a table.

Step 13. The next step is to identify transcripts which have at least one significant probeset based on the p-values and fold changes computed above. Click on the *Transcripts with Significant Probesets* link and then select p-value cut-off of 0.01 and a fold change cut off of 1.5. This will select only probesets with these properties. A new dataset is created in the navigator which has these probesets; this dataset also includes all probesets which belong to the same transcripts as the selected probesets.

Step 14. Now we have a set of transcripts which has at least one significant probeset. Transcript signal values for these transcripts can be obtained by clicking on the *Transcript Summarization* link in the *Splicing Analysis* section of the workflow browser. This will create a new dataset called the *Splicing Analysis Dataset* whose columns contain both probeset and transcript signals.

Step 15. Now that we have both probeset signals and transcript signals for transcripts which have at least one significant probeset, we can identify transcripts which are significantly differentially expressed and transcripts which show significant splicing, i.e., some probesets/exons in these have signal values which differ substantially from the transcript signal values. The first of these steps can be performed by clicking on the Significance Analysis Wizard in the *Transcript Significance Analysis* subsection of the workflow browser. Do the same on this wizard as in Step 13. This will compute p-values and fold changes via a paired T-Test for each transcript.

Step 16. Create a gene list of significant transcripts. First, select a cell on the Differential Expression report view which corresponds to p-value less than 0.05 and fold change greater than 1.5. Then click on the *Create Probeset List* link in the workflow browser. Give this list a name (say "transcripts-sig") and specify Transcipt Cluster Id as the id of interest. The *GeneList* section on the bottom left of **ArrayAssist** should now show this new gene list.

Step 17. Next, we identify transcripts which show significant splicing, i.e., some probesets/exons in these have signal values which differ substantially

Test Description

Test name:	T Test paired
Pvalue computation:	Asymptotic
Correction type:	Benjamini-Hochberg

Result Summary

[Tumor] Vs [Normal]

	P all	P < 0.05	P < 0.02	P < 0...	P < 0....	P < 0....
FC all	369	291	184	1	0	0
FC > 1.1	338	286	181	1	0	0
FC > 1.5	143	140	112	1	0	0
FC > 2.0	30	30	25	0	0	0
FC > 3.0	7	7	5	0	0	0
Expected...		18	7	3	1	0

Figure 6.9: Selecting Significant Transcripts

from the transcript signal values. To do this, click on the Splicing Analysis Dataset in the navigator and then on the Significance Analysis Wizard in the *Splicing Significance Analysis* subsection of the workflow browser. Do the same on this wizard as in Step 13. This performs a paired T-Test on the log-scale splicing indices (i.e., the difference between the log-scale probeset and the log-scale transcript signals). This test results in p-values and fold changes between the normal and tumor groups for each probeset. A fold change of 2 for a probeset means that the linear-scale splicing index goes up by a factor of 2 between normal and tumors.

Step 18. Create a gene list of significantly spliced transcripts. First, select a cell on the Differential Expression report view which corresponds to p-value less than 0.05 and fold change more than 1.5. Then click on the *Create Probeset List* link in the workflow browser. Give this list a name (say "splice-sig") and specify Transcipt Cluster Id as the id of interest. The *GeneList* section on the bottom left of **ArrayAssist** should now show this new gene list.

Step 19. Move to the Splicing Analysis Dataset in the navigator and then select the two gene lists created above in the *GeneList* section on the bottom left of **ArrayAssist** . Then right-click, and invoke a Venn Diagram. This will show transcript counts of transcripts that are differentially expressed and/or differentially spliced across experimental groups.

Step 20. Next, we will create 3 sub-datasets of the Splicing Analysis Dataset, one corresponding to transcripts which are differentially spliced but not differentially expressed, another corresponding to transcripts that are differentially expressed but not spliced. and yet another corresponding to transcripts that both differentially spliced and expressed. To do this, first select the appropriate region on the venn diagram and then use the *Create New Subset from Selection* operation on the *Data* menu. This will create a new child dataset of the Splicing Analysis Dataset. Remember to move to the Splicing Analysis Dataset each time to create a data subset.

Step 21. Now we visually explore the subsets created, in particular the dataset corrsponding to transcripts which are differentially spliced but not differentially expressed. Move to this dataset in the navigator and click on the *Differential Transcript vs Differential Splicing* view in the *Splicing Views* section of the workflow browser. Select the "TissueType" checkbox and on the next page, select the first group as Tumor and the second as Normal This creates a scatter plot in which probesets cosrresponding to a particular transcript appear as a single straight horizontal line. Low transcript

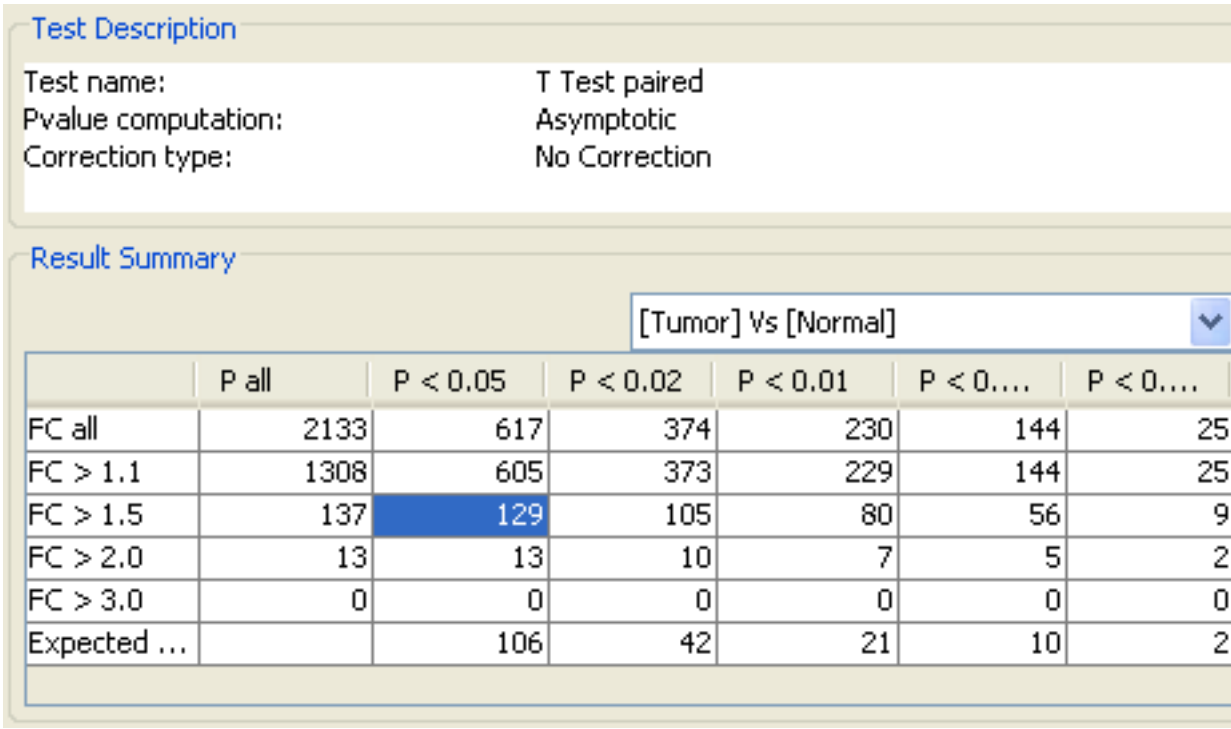


Figure 6.10: Selecting Significantly Spliced Transcripts

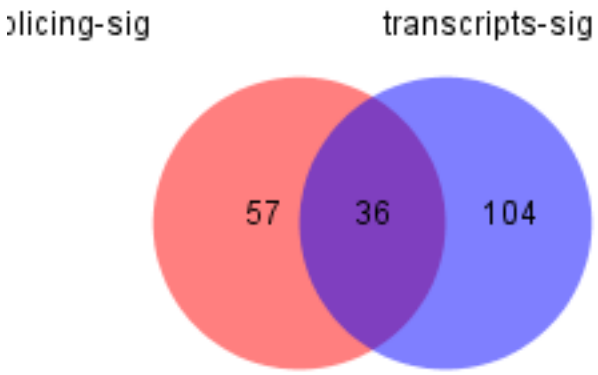


Figure 6.11: Venn Diagram

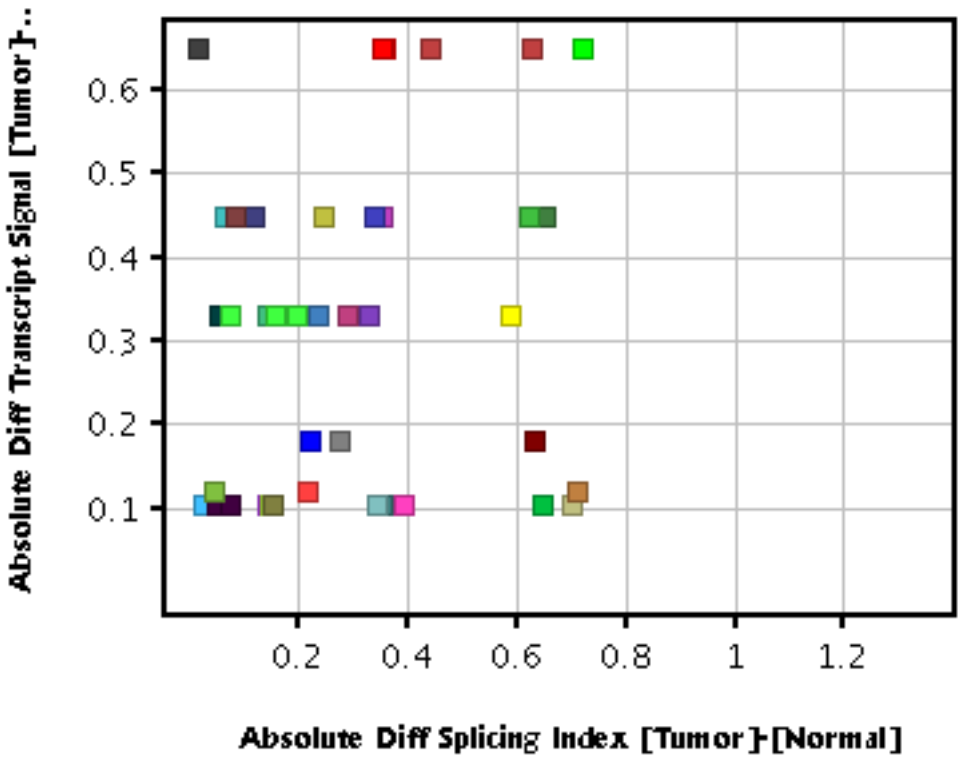


Figure 6.12: The Differential Transcript vs Differential Splicing View

differential expression means that these horizontal lines appear close to the x axis. High splicing differentials mean that these horizontal lines stretch out to the far right. Note that both x and y axes are absolute values.

In particular, note that the exon represented by the yellow dot in the transcript which lies in the middle of the plot seems to behave differently from the remaining exons in that transcript. Select this dot and see the splicing differential analysis Volcano plot, this exon has a very low p-value for splicing, indicating significant differential splicing.

Step 22. Also click on the *Differential Splicing Index along Chromosome* view in the workflow browser and provide the same choices. Use the *Tile Both* option from the *Windows* menu to tile all the windows. Note that this view is segregated by chromosome and you can move across chromosomes

using the chromosome dropdown. Each probeset is plotted on this view on the appropriate chromosome at a y-coordinate that depends on its splicing index. The points in this view are colored by exons, so probesets on the same exon appear in the same color.

Step 23. You can zoom in on any of the two views by right-clicking on that view and choosing zoom-mode. You can also select points on any of the two views by right-clicking on that view and choosing select-mode and then dragging a rectangle around the required points. Select a single full transcript on the *Differential Transcript vs Differential Splicing* view (zoom in prior to selection, if necessary); this transcript will also be selected in the *Differential Splicing Index across Chromosome* view automatically due to dynamic linking. To locate this transcript on the latter view, use the dropdown to browse through the chromosomes until you see a mass of yellow points, then zoom into these points and Right-Click and clear selection. This will show you how the probesets/exons in this transcript appear along the chromosome. One or more exons appering together on the chromosome and showing splicing indices distinct from the other exons indicate differential splicing phenomena at play between the normal and the tumor samples.

When we zoom into the transcript on interest which we identified in the previous step, the yellow exon again seems to behave substantially differently than the rest.

Step 24. Select all probesets in the interesting transcript above. Then click on the *Profile Plot: Splicing Index* link in the *Splicing Views* section of the workflow browser. Select the "TissueType" checkbox and on the next page, select the first group as Tumor and the second as Normal. This will show a profile plot of splicing indices; the differential splicing pattern of the interesting exon (colored blue) over groups should be visually apparent in this view. Adjust the properties on the view using the Right-Click Properties dialog if necessary.

Step 25. To see annotations for this interesting transcript and probesets above, click on the *Import Annotations* link in the *Utilities* section of the workflow browser and choose the Refseq, Genbank, Gene Symbol columns; these will be imported into the current dataset. With the interesting probesets selected, open the Lasso view from the *View*$\longrightarrow$*Lasso* menu item and then customize the columns on this view by using *Right-Click* $\longrightarrow$*Properties*$\longrightarrow$*Columns* so these newly imported columns are present. Now click on any of the annotation columns of interest and it will take you to the appropriate web site for more details on this.

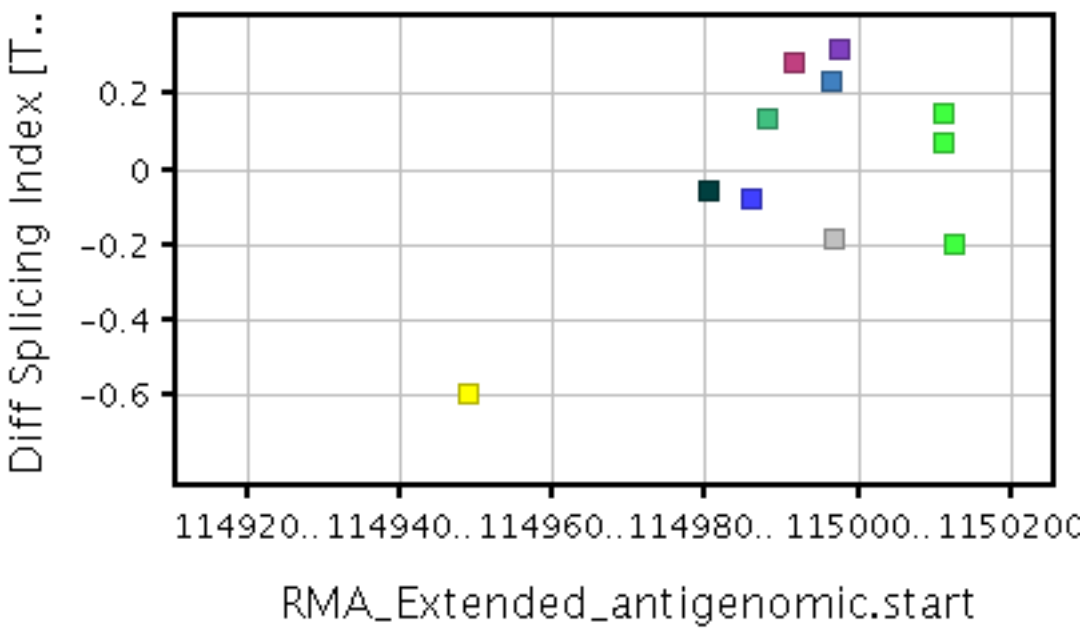


Figure 6.13: A transcipt showing potential splice variation effects in the Differential Splicing Index along Chromosome View

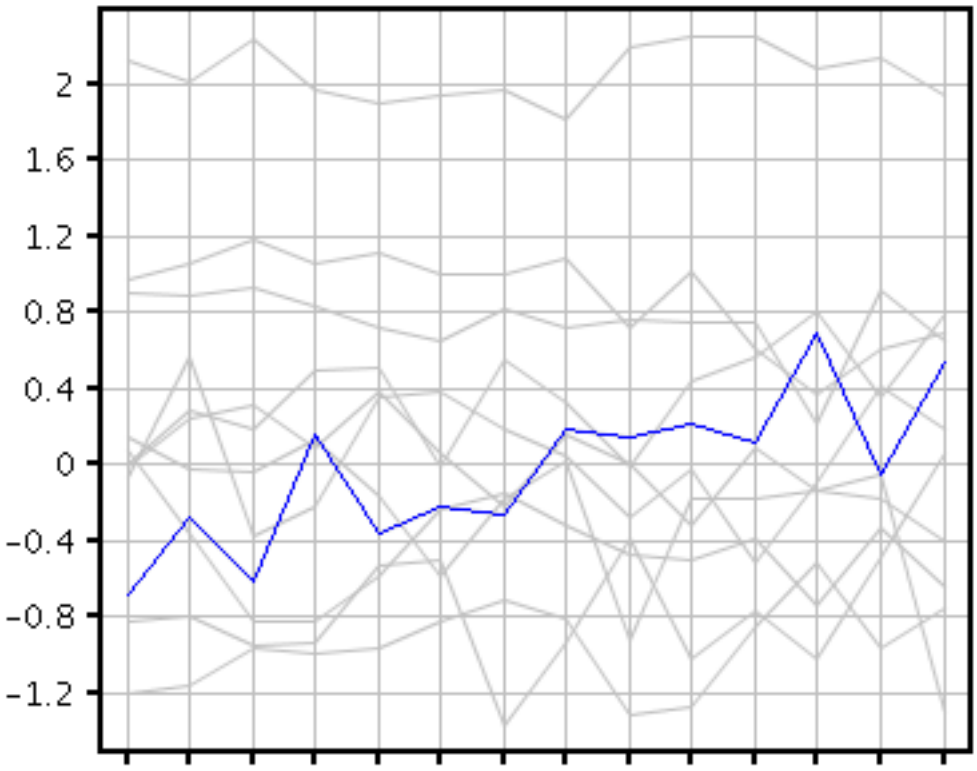


Figure 6.14: A transcript showing potential splice variation effects in the Profile Plot Splicing Indices view

Step 26. You can also view the interesting transcript selected above in the contect of the genome browser. Launch the Genome Browser from the corresponding link on the workflow browser. Then click on the Add Tracks  icon on the genome browser window. Add KnownGenes static track by selecting them and clicking on the AddTrack button. Also add the data track corresponding to the current dataset. Then click on the NextSelected  icon; this will focus the genome browser so the selected probesets are right at the center. Now zoom into the relevant region by repeatedly clicking on the zoom  icon. The chromosomal area around the probesets of interest can not be seen here. You can scroll left or right using the arrows at the bottom- right and bottom-left respectively. Click on the data track name corresponding to the current dataset and height this track by the differential splicing index (which can be obtained by clicking on the Differential Splicing Index link in the Utilities section of the workflow browser). The exon of interest stands out again.

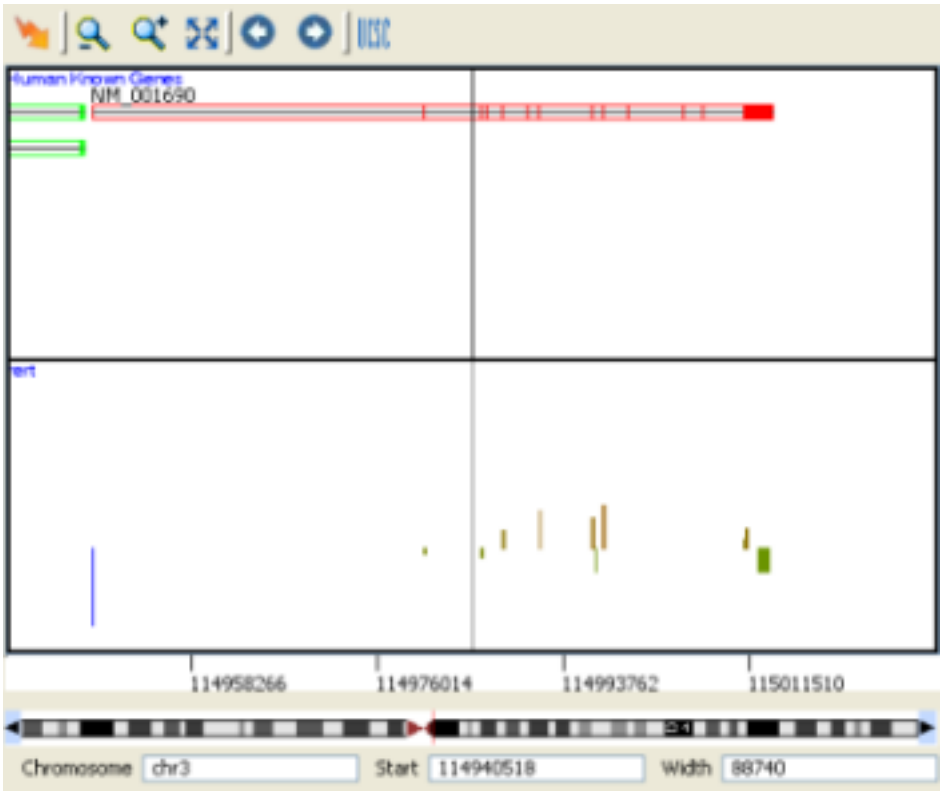


Figure 6.15: Region around potentially alternatively spliced probeset

Chapter 7

Importing Copy Number Data

7.1 Importing Genotyping Data for Copy Number Analysis

Use the following command to import CEL files into **ArrayAssist** to create a new Copy Number project.

File⟶New Affymetrix Copy Number Project

NOTE: Affymetrix CEL and CHP files are available in two formats, the Affymetrix GeneChip Command Console compliant data file (AGCC) files; and Extreme Data Access compliant data (GCOS
XDA) files. **ArrayAssist** 5.1 uses the recently released Affymetrix Fusion SDKs that supports both AGCC and XDA format CEL and CHP files. However the older Affymetrix GDAC SDKs are also avaliable in **ArrayAssist**. By default, **ArrayAssist** uses the GDAC SDKs. The Fusion SDKs can be used by changing the defult settings in *Tools ⟶Options ⟶Affymetrix Probe-Level Analysis ⟶Fusion*

7.1.1 Selecting CEL Files

The first step in creating the project is to provide a project name and folder path and then select CEL files of interest. The project folder will be used to save the .avp project file in addition to several pieces of intermediate information created while processing CEL files.

To select files, click on the *Choose File(s)* button, navigate to the appropriate folder and select the files of interest. Use Left-Click to select the first file, Ctrl-Left-Click to select subsequent files, and Shift-Left-Click for a contiguous set of files. Once the files are selected, click on *OK*. If you wish to select files from multiple directories or multiple contiguous chunks of files from the same directory, you can repeat the above exercise multiple times, each time adding one chunk of files to the selection window. You can remove already chosen files by first selecting them (using Left-Click , Ctrl-Left-Click and Shift-Left-Click , as above) and then clicking on the *Remove Files* button. After you have chosen the right files, hit the *Next* button. Note that the dataset will be created with each column corresponding to one CEL file or one experiment.

NOTE: The order of the columns in the dataset will be the same as the order in which they occur in the selection interface. If you want the columns in the dataset to be in any specific order, you should order them here appropriately.

Both the 100K arrays and the 500K arrays currently comprise two actual arrays of half the size each (the 100K arrays have Xba and Hind arrays of size 50K each and the 500K arrays have NSP and STY arrays of size 250K each). **ArrayAssist** will attempt to automatically pair up the arrays based on naming rules. However, this pairing can be modified on the next page if required. Note that **ArrayAssist** allows partial pairs, i.e., you can specify one or both CEL files for each pair when creating your project. Data from paired CEL files will be automatically combined and presented in one column in **ArrayAssist** . If only one of the two CEL files in a pair is provided, then the data values corresponding to the other array in the pair will be represented as missing (unless, for instance, only Xba CEL files are provided, in which case, all data columns will be restricted to just Xba probesets).

NOTE: The disk space required per 100K CEL file is approximately 40-50MB. If the required amount of space in not available, CEL file processing could abort midway.

7.1.2 Getting Chip Information Packages

To import Genotyping CEL files, you will need *Chip Information Packages* for your chips of interest. These packages contains probe layout information

derived from the CDF file as well as SNP annotation information derived from the NetAffx comma separated annotation file. You can fetch this file using *Tools⟶Update Data Library*.

NOTE: **Chip Information Packages** could change every quarter as new gene annotations are released on NetAffx by Affymetrix. These will be put up on the **ArrayAssist** update server. **ArrayAssist** will directly keep track of the latest version available on **ArrayAssist** update server. When **ArrayAssist** launches, it will check the version available on the local machine with the version on the server. If a newer version has been deployed on the server, then, on starting, **ArrayAssist** will launch the update utility with the specific libraries check and marked for update.
Each project stores the generation date of the *Chip Information Package*. If newer libraries are available on the tool, when the project is opened, you will be prompted with a dialog asking you whether you want to refresh the annotations. Clicking on *OK* will update all the annotations columns in the project. You can also refresh the annotations after the project is loaded from the *Refresh Annotations* link in the workflow.

7.2 Running the Copy Number Workflow

When the new Affymetrix Copy Number project is created after proceeding through the above *File⟶New Affymetrix Copy Number Project* wizard, **ArrayAssist** with open a new project with the following view:

The Data Description View: This view shows a list of CEL files imported in the panel on the left. The *File Header* tab shows the file header containing some statistics for the file selected on the left panel.

You are now ready to run the Affymetrix Copy Number Workflow. The Affymetrix Copy Number Workflow Browser contains all typical steps used in Copy Number analysis. These steps will output various datasets and views. The following note will be useful in exploring these views.

NOTE: Most datasets and views in **ArrayAssist** are *lassoed*, i.e., selecting one or more rows/columns/points will highlight the corresponding rows/columns/points in all other datasets and views. In addition, if you select probesets from any dataset or view, signal values and gene annotations for the selected probesets can be viewed using *View* $\longrightarrow$*Lasso* (you may need to customize the columns visible on the Lasso view using Right-Click *Properties*).

7.2.1 Providing Experiment Grouping Information

Experiment Factors and Groups. Click on the *Experiment Grouping* link in the workflow browser. The *Experiment Grouping* view which comes up will initially just have the CEL file names (CEL file pairs are paired up and represented as a single unit). The task of grouping will involve providing more columns to this view containing *Experiment Factor* and *Experiment Grouping* information. A Control vs. Treatment type experiment will have a single factor comprising 2 groups, Control and Treatment. A more complicated Two-Way experiment could feature two experiment factors, genotype and dosage, with genotype having transgenic and non-transgenic groups, and dosage having 5, 10, and 50mg groups. Adding, removing and editing experiment factors and associated groups can be performed using the icons described below.

Reading Factor and Grouping Information from Files. Click on the Read Experiment Grouping from File  icon to read in all the Experiment Factor and Grouping information from a tab or comma separated text file. The file should contain a column containing CEL/CHP file names; in addition, it should have one column per factor containing the grouping information for that factor. Here is an example tab separated file. The result of reading this tab file in is the new columns corresponding to each factor in the Experiment Grouping view.

```
#comments
#comments
filename genotype dosage
A1.CEL    NT       0
A2.CEL     T       0
A3.CEL    NT      20
A4.CEL     T      20
A5.CEL    NT      50
A6.CEL     T      50
```

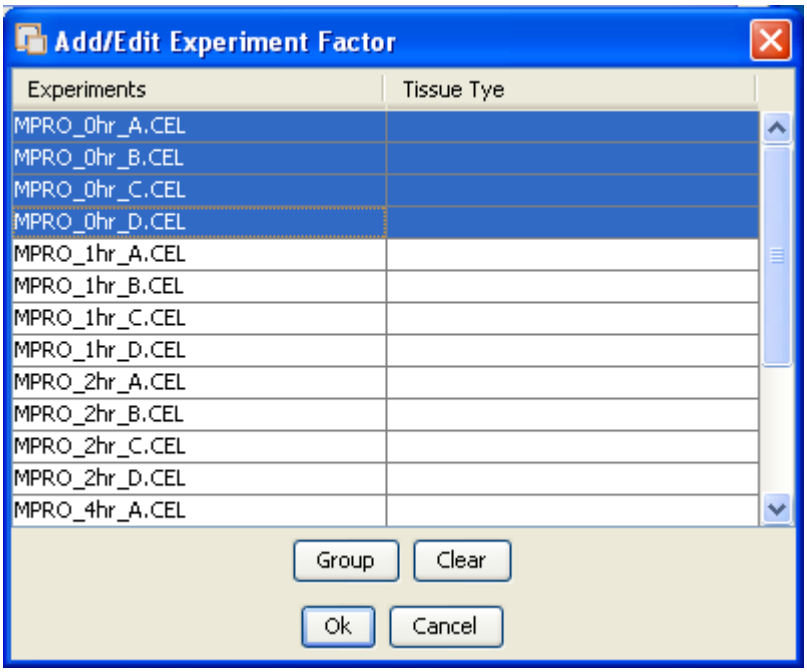


Figure 7.1: Specify Groups within an Experiment Factor

Adding a New Experiment Factor. Click on the Add Experiment Factor icon to create a new experiment factor and give it a name when prompted. This will show the following view asking for grouping information corresponding to the experiment factor at hand. The CEL/CHP files shown in this view need to be grouped into groups comprising biological replicate arrays. To do this grouping, select a set of CEL/CHP files, then click on the *Group* button, and provide a name for the group. Selecting CEL/CHP files use Left-Click , Ctrl-Left-Click , and Shift-Left-Click , as before.

Editing an Experiment Factor. Select the experiment factor you want to edit, by clicking on the respective factor column. This column will be selected. Click on the *Edit Experiment Factor* icon to edit an Experiment Factor. This will pull up the same grouping interface described in the previous paragraph. The groups already set here can be changed on this page.

Remove an Experiment Factor. Click on the Remove Experiment Factor

 icon to remove an Experiment Factor.

7.2.2 Generating Genotype Calls

Currently **ArrayAssist** supports two ways of incorporating Genotype Calls; the first is by importing calls from CHP files, and the second is by generating calls using built-in algorithm (the latter is not yet implemented and will be available in a future version). The calls output are AA and BB (homozygous) , AB (heterozygous) or No Call (the algorithm is unable to determine the call with sufficient confidence).

Importing Calls from CHP files requires providing the CHP file names. These names should differ from the corresponding CEL file names only in file extension. A new dataset is then created with the imported Genotype Calls.

Once implemented, clicking on the *Generate Genotype Calls* link will use the BRLMM algorithm to generate calls. However, for BRLMM to run, the number of arrays hasto be more than 6. A new dataset will then be created with the imported GenotyeCalls. For more details on BRLMM, see Technical section below.

7.2.3 Reference Creation

ArrayAssist supports both analysis with and without paired normal samples. Analysis without paired normal samples is performed by comparing against reference samples. One reference set is prepackaged with **ArrayAssist**. However, if you wish to create your own reference sample set, you can do so using the *Create Reference* link.

To create a new reference, first select the experiment group (if you wish to create a reference out of all the CEL files in the project then you will need to create a new factor in the Experimental Grouping View and give all CEL files the same group name; see Experiment Grouping, and then specify which of the arrays chosen has male gender. You need to ensure that the the dataset currently in focus is a genotype calls dataset.

The reference creation process will generate signals for each of the CEL files chosen. The signals are then averaged and stored as part of the reference files (along with their standard deviations). The aim of specifying genders for the CEL files is to perform adjustments on X chromosome signals; the average X chromosome signals for males are equalized to the average X chromosome signals for females via scaling the male signals; here the average is taken over all arrays with the corresponding gender and over all SNPs on

Chromosome X. So effectively, the reference stores a female signal. Additionally, genotype calls will be picked up from the current dataset in focus and various statistics on the genotype calls needed to perform Loss of heterozygosity (LOH) and copy number analysis against the reference are also computed and stored in the reference file. See Technical Section for more details on these quantities.

The reference created is stored in a .cnr file. Any of these .cnr reference files can then be used in the Copy Number Analysis against Reference link.

Finally, note that precreated reference files for both the 100K and the 500K arrays are prepackaged with the chip library package. These references are located in the app/DataLibrary/GenoChip subfolder of the **ArrayAssist** installation directory. For instance, the reference file for Xba 50K arrays is
app/DataLibrary/GenoChip/Mapping50K_Xba240/Chip/Reference.cnr
and the reference file for Xba+Hind combined 100K arrays is at
app/DataLibrary/GenoChip/Mapping50K_Xba240/Chip/CombinedReference.cnr
app/DataLibrary/GenoChip/Mapping50K_Hind240/Chip/CombinedReference.cnr.

7.2.4 Copy Number and LOH Computation

ArrayAssist supports both analysis with and without paired normal samples. To run this analysis, the current dataset in the navigator must be the Genotype Calls dataset obtained as described in Genotype Calls.

Analysis without Paired Normals. Analysis without paired normal samples is performed by comparing against reference samples. Precreated references are prepackaged with the library package for the relevant chip. These references are located in the app/DataLibrary/GenoChip subfolder of the **ArrayAssist** installation directory. For instance, the reference file for Xba 50K arrays is
app/DataLibrary/GenoChip/Mapping50K_Xba240/Chip/Reference.cnr
and the reference file for Xba+Hind combined 100K arrays is at
app/DataLibrary/GenoChip/Mapping50K_Xba240/Chip/CombinedReference.cnr
app/DataLibrary/GenoChip/Mapping50K_Hind240/Chip/CombinedReference.cnr.
References for 50/100K arrays are derived from 90 CEL file pairs obtained from http://www.affymetrix.com/support/technical/sample_data/hapmap_trio_data.affx and references for 250/500K arrays are derived from 40 CEL file pairs obtained from http://www.affymetrix.com/support/technical/sample_data/500k_data.affx. These references are gender corrected as

described in Create Reference. You can create custom Reference files on your CEL files as well as described in Section Create Reference.

Click on the *Analysis against Reference* link in the wokflow browser. Provide the name of the appropriate .cnr reference file you wish to compare against. Also provide the experiment group which you wish to generate copy numbers and LOH scores for. If you wish to do this for all CEL files in the project then you will need to create a new factor in the Experimental Grouping View and give all CEL files the same group name; see Section on Experiment Grouping.

This operation creates a new dataset with the following information. First, log ratios (signals for each array divided by signals in the reference file, and then log transformed) are computed for each selected array. Second, an Hidden Markov Model is used to convert signal values to inferred copy number estimates (values 1,1.5,2,2.5,3,4). Finally, another Hidden Markov Model is used to infer LOH scores (between 0 and 1, higher scores are more significant) from genotype calls. See Technical Details for more details on each of these algorithms.

Paired Normal Analyis. Click on the *Paired Normal Analysis* link in the wokflow browser. Provide the two experiment groups which you wish to compare. Typically you will chose two groups, though in general, more than two groups could be chosen, and pairs amongst these compared. On the next page, adjust the order of arrays in each group so the arrays are properly paired. The next page will show a list of pairs of all groups selected; typically if you have chosen only two groups, only one pair would appear. Select the pairs of interest and then order order each pair so that the normal or control is group2 and the treatment or disease tissue is group1.

This operation creates a new dataset with the following information. First, log ratios (signals for each array divided by signals in the corresponding normal, and then logged) are computed for each selected array. Second, an Hidden Markov Model is used to convert signal values to inferred copy number estimates (values 1,1.5,2,2.5,3,4) relative to the normal signals. Finally, another Hidden Markov Model is used to convert signal values to LOH scores (between 0 and 1, higher scores are more significant) from genotype calls of disease and normal tissue.

See Technical Details for more details on each of these algorithms.

Importing from CNAT. In addition to running algorithms within **ArrayAssist** , you also have the option of importing copy number and LOH information data from CNAT output. You will need the .cnt files output by CNAT for each of the arrays imported in the project. Specify the .cnt

file names; log ratios, copy numbers (the GSA_CN columns), copy number p-values (which are presented on the log base 10 scale with a negative sign in case the log ratio is negative) and LOH scores (which are again negative log base 10 of probability of LOH) are imported in.

7.2.5 Identify Regions/Genes

Once copy number values and LOH scores have been generated, the next step is to identify genomic regions which have a significant copy number value or LOH score, and then to identify genes which are in these region.

Identify Significant Regions. This dialog asks you to specify a region length s, a SNP percentage f, and minimum number of arrays t. In addition, it asks for specifying conditions on copy numbers, LOH scores, log ratios etc; select the quantities of interest and specify the appropriate thresholds. This information is now processed as follows.

First, for each array and each region of length s, the fraction of SNPs in this region which satisfy all of the conditions specified is calculated for this array. If this fraction is greater than f, and this holds for at least t arrays, then all SNPs in this region are selected. All selected SNPs are aggregated into a new dataset.

The significance condition is obtained by taking a conjunction of all selected conditions (i.e., all selected conditions have to be true). Selected conditions can be specified on absolute calls (AA,BB,AB,No Call), copy number (1,1.5,2,2.5,3,4), LOH scores (between 0 and 1, higher scores are more significant) and signal log-ratios. In addition, conditions can be specified on columns imported from CNAT output, i.e., copy number ($>= 0$), copy number p-values (which are actually on the log base 10 scale with positive values corresponding to positive log ratios and negative values corresponding to negative log ratios) and LOH scores ($>= 0$, higher the better). Thus, filtering can be done simultaneously on the Copy Number and the Copy Number p-value.

There is also an option here to select just individual SNPs and not regions. SNPs which satisfy all specified conditions in at least t arrays are selected. All selected SNPs are aggregated into a new dataset.

It is also possible to search for a SNP in a specific gene or the cytoband region. In the parent spreadsheet, using the Import Annotations function, from the workflow on the right, import the associated gene gene Id column. Go to *Annotations* $\longrightarrow$*Search Genes*. Specify the desired columns and the keyword you want. This selects the rows which contain the SNPs in the named gene. In order to run a search using the cytoband. Note

that if the cytoband is 1q23.3, then the cytoband column contains q23.3 and this can be used as the keyword. The search can be further restricted to chromosome 1 by using the Filter, present near the workflow. **Identify Significant Genes.** Select any subset of SNPs from the current dataset (since all datasets are lassoed, you could select SNPs from any other dataset or from the genome browser and then move to the current dataset in the navigator).

Clicking on this link will create a spreadsheet of HG-U133Plus_2 probesets which have either endpoints within genomic upstream distance *ul* or downstream distance *dl* of any of the selected SNPs. The *ul* and *dl* values are configurable via *Tools*⟶*Options*⟶*CopyNumber* ⟶*Gene Overlap Region Settings*.

NOTE: As you explore significant SNPs/Regions either via the genome browser or via one of these above filtering methods, you might want to label and track SNPs which are significant. Use *Data*⟶*Row Commands* ⟶*Label Rows* to add another marker column to you current dataset. All selected SNPs will get the specified label in the specified column. You can keep adding new labels to the same column, thus adding to the list of labelled SNPS.

7.2.6 Import Annotations

SNP annotations available in NetAffx are packaged with the library packages and can be imported into the currently open dataset via this link.

7.2.7 Genome Browser

The Genome Browser can be invoked using this link. This browser allows viewing of several static prepackaged tracks, data tracks based on data in currently open datasets, and profile tracks based on data in currently open datasets. For more details on usage, see Section on Genome Browser. Profile tracks are the most useful for viewing copy number and LOH data, as shown in the image below.

7.2.8 Space Requirements

Please note the following special requirements for working with genotyping CEL files which contain much larger amounts of data than the largest Affymetrix 3'IVT chips.

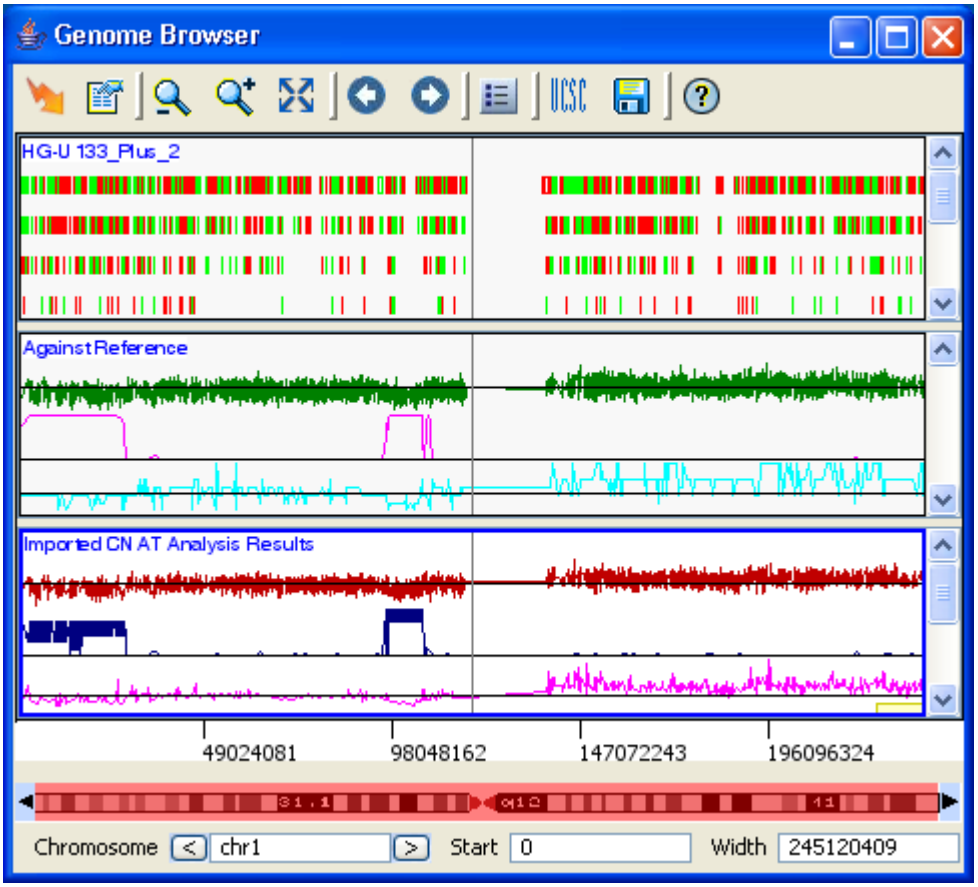


Figure 7.2: Profile Tracks in the Genome Browser

Disk Space Requirement. Please make sure that the amount of disk space available is at least 40-50MB per 100K CEL file you wish to process. This space must be available on the disk drive in which your project is being saved. Probset summarization will stop midway if this amount of space is not available.

Memory Setup. It is recommended that you have a 2GB RAM machine for processing Genotyping files. It is also recommended that you make the following modification in the installation-folder/bin/packages/properties.txt file which can be edited using Wordpad or any other text editor: in the java.options line, modify -Xmx1024m to -Xmx1500m. Shut down **ArrayAssist** before making this change and relaunch after the change is made for the change to take effect. This change allows Java to use a larger amount of memory on your machine.

Note that on some machines, launching **ArrayAssist** after making this change will cause all text to blank out; in such cases, you will need to set your hardware acceleration configuration on your machine (on Windows XP, go to *My Computer* $\longrightarrow$*Display* $\longrightarrow$*Settings* $\longrightarrow$*Advanced* $\longrightarrow$*Troubleshoot* and set the acceleration to the third bar from the left).

In addition, on some rare machines, **ArrayAssist** will not start up at all with the above change. The reason for this is the presence of some other applications having reserved certain memory slots. In such a situation, the best course of action would be to reduce the -Xmx value above to a lower value. You will need to identify the highest value for which **ArrayAssist** starts up via trial and error. This will affect the number of CEL files that can be processed in one project. Alternatively, use a fresh new machine without other applications installed.

Memory Requirement. ArrayAssist has been optimized to import in and generate signal-log ratios, LOH scores, Copy Numbers and Genotype Calls for about 100 500K arrays at a time on a 2GB Windows machine.

Keeping Track of Memory Usage. Finally, keep a watch on the memory monitor at the bottom right of **ArrayAssist** , which shows a message stating that the application is using x MB of y. Click on the garbage can icon at the bottom right occassionally to force **ArrayAssist** to release memory. If y starts getting close to the limit specified in -Xmx option above then make sure you save your project and delete the main probeset summarized dataset, keeping only the splicing analysis dataset and all children datasets thereof. This will provide plenty of memory for further downstream operations. An operation that demands a large amount of memory causing

application memory to cross the -Xmx limit set above could cause an application crash.

7.2.9 Algorithm Technical Details

Signals. Signal Generation is performed by using Quantile Normalization followed by running RMA twice, once each on the A and B alleles; these are the allele specific signals. The combined signal is the average of these two signals. This step is identical to the signal generation step of the BRLMM genotype calling algorithm.

Calls. Once the BRLMM algorithm is made available in **ArrayAssist** , Genotype calls will generated using the DM algorithm if the number of arrays is less than 6 and using the BRLMM algorithm when the number of arrays is greater than 6.

Log Ratios. Log ratios are computed by taking ratios of signals on the current array and the signals in either the paired normal or the reference .cnr file, and then logging to base 2.

Copy Number Hidden Markov Model (HMM). Copy numbers for both paired normal analysis and analysis against a reference are generated from signals using an HMM very similar to the one described in the dChip paper http://www.broad.mit.edu/mpr/publications/projects/SNP_Analysis/Zhao_2004.pdf. It has 6 states, corresponding to copy numbers 1, 1.5, 2, 2.5, 3 and 4 respectively. Emission probabilities at state j for SNP i are assumed to be normally distributed with mean μ_{ij} and deviation σ_{ij}, where μ_{ij} equals $j/2$ times the average signal for SNP i in the paired normal or in the reference, and σ_{ij} is the standard deviation of SNP i in the reference (in the case of paired normal analysis, σ_{ij} is picked up from the pre-stored reference). Transition probablities and initial probabilities are exactly as in http://www.broad.mit.edu/mpr/publications/projects/SNP_Analysis/Zhao_2004.pdf.

LOH Analysis against Reference Hidden Markov Model. LOH scores for analysis against a reference are generated from genotype calls using an HMM with 3 states, representing Loss of Heterozygosity (L), Retention of heterozygosity (R-HET), and Retention of Homozygosity (R-HOM), respectively. The emission probabilities at L and R-HOM are set to .99 for Homozygous and 0.01 for Heterozygous. The emission probabilities at R-Het are set to .99 for Heterozygous and 0.01 for Homozygous. Transition probabilities are defined exactly as in http://galton.uchicago.edu/~loman/

To↓, From→	LOH	Hom-R	Het-R
LOH	(1-θ)+θ P_o(L)	θP_o(L)	θP_o(L)
Hom-R	θP_o(R)P(Hom)	[(1-θ+θP_o(R)]P_R(Hom|Hom)	[(1-θ)+θP_o(R)]P_R(Hom|Het)
Het-R	θP_o(R)P(Het)	[(1-θ+θP_o(R)]P_R(Het|Hom)	[(1-θ)+θP_o(R)]P_R(Het|Het)

Figure 7.3: Transition Probabilities for LOH analysis againt Reference HMM

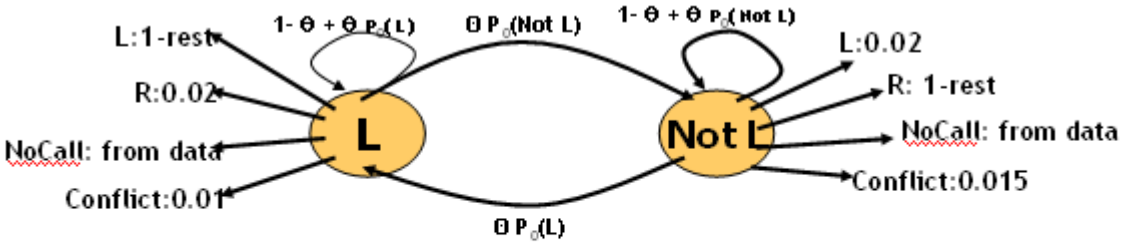
`thesis/Thesis_double.pdf` and very similar to the dChip paper http://compbiol.plosjournals.org/perlserv/?request=get-document\&doi=10.1371/journal.pcbi.0020041 and are recapitulated in the image below.

Here, $P_0(L) = .01$, $P_0(R) = 0.99$ and θ is set to $1 - e^{-2d}$ where d is the distance between the current and previous SNPs in units of 100MB. Note that $P_0(L)$ can be modified to a user defined value between 0 and 1 via *Tools*$\longrightarrow$*Options* $\longrightarrow$*CopyNumber* $\longrightarrow$*LOH HMM*. A higher value would increase the number of LOH regions detected but also increase false positives.

For analysis against reference, all the probabilities mentioned in this image above are computed from reference CEL files and stored in the .cnr reference file.

For paired normal analysis, a different (simpler) HMM shown in the following figure is used; the emission alphabet is no longer genotype calls but a Loss, Retention, Conflict or Non-Informative call computed from the paired samples as indicated in the figure. The starting probability of loss defaults to 0.01 and can be set via *Tools*$\longrightarrow$*Options* $\longrightarrow$*LOH HMM*. A smaller value would lead to fewer LOH calls.

Note that the L,C,N,R calls are not explicitly output in the spreadsheet; these can be obtained via a custom script; contact support to request a custom script.



		Tumor			
		A	B	AB	No Call
Normal	A	N	N	C	N
	B	N	N	C	N
	AB	L	L	R	N
	No Call	N	N	R	N

Figure 7.4: The Paired Normal HMM

Chapter 8

Analyzing Single-Dye Data

ArrayAssist can access and analyze files obtained by image analysis of most Single-Dye array formats with the following properties.

- There is usually one data file per experiment containing all spot quantified data for that experiment.
- The actual spot data in the data file is in tabular form, i.e., it is laid out as rows and columns, typically one row per spot with columns corresponding to various spot properties like gene name, block location, subblock location, foreground mean/median intensity, background mean/median intensity, etc.
- The tabular portion of the file could be only a part of the file and could be preceded by several lines containing additional experiment annotation details and possibly followed by several such lines as well.

Import of single-dye array formats happens via the two step process below.

Create Import Template. First, you need an *Import Template* for the specific files of your interest. **ArrayAssist** comes prepackaged with templates for the following file formats:

- **abi:** Standard abi files in a plain text format containing only data.
- **abi_multi:** ABI files where all the experiments are output into a single file.
- **ABI_1700:** ABI files output from a standard ABI1700 version.

- **codelinkV3-5:** CodeLink Expression Analysis software (versions 3 through 5) output formats.
- **combimatrix:** Standard Combimatrix single dye template.
- **illumina_probe_profile:** Template for files generated from Illumina Inc. BeadStudio version 2.3.4
- **illumina_gene_profile:** Template for files generated from Illumina Inc. BeadStudio version 2.3.4

If you are working with one of these formats, try the appropriate template first by going through the *File* $\longrightarrow$*New Single-Dye Project* wizard. If it does not work (which might happen because of version differences) or if you are working with some other format, then you have two choices.

- Build your own template. This can be done for most formats which have data corresponding to one experiment in each file. See the description in Section The Single Dye Import Wizard for details.
- Seek **ArrayAssist** support for building the template. Send mail to `techservices@stratagene.com`and provide two sample files which you wish to import. We will send you a new template which will enable you to import your files into **ArrayAssist**.

Note that you *cannot* build your own templates where all the experiments are output into a single file. In such situations, if you could provide a sample file, we will be able to build a templete to import such files. We have included a template abi_multi where the output file contains many experiments.

Run Analysis. Second, import the files using this template and use the menu and workflow browser operations to proceed with the analysis. To perform the import, use the *File* $\longrightarrow$*New Single-Dye Project*. This will launch a wizard; choose the files of interest and provide the template name. See Section The Single Dye Workflow for details on further analysis.

8.1 The Single Dye Import Wizard

Step 1 - Select Files Use the *Choose File(s)* option on the wizard to locate the files of interest. Use this multiple times to locate files from

different locations. *Remove file(s)* option can be used to remove selected files.

The Separator separates fields in the file to be imported and is usually a tab, comma or space; new separators can be defined by scrolling down to EnterNew and providing the appropriate symbol in the textbox.

The Text Indicator is usually just inverted commas (") used to ignore separators which appear within text strings.

The Missing Value Indicator indicates the symbol(s), if any, used to represent a missing value in the file. This applies only to cases where the value is represented explicitly by a symbol such as N/A, NA or —.

Comment Indicators are markers at the beginning of the line which indicate that the line should be skipped (typical examples is the # symbol).

Step 2 - Select Template Use the *Select a template* drop down menu option to check if the format of interest is prepackaged. If not, use the *None* option and use the easy template building steps to create a template for the data. The template can be then saved. This template once created will become part of the drop down menu option and will be available from the next time.

Step 3 - Format Options Use this step to specify the exact format of the data being brought in. Use the *Separator* option to specify the type of file. Use the *Text qualifier* to specify any special qualifiers used in the data file. Similarly use the *Missing value indicator* and *Comment indicator* to define the format of the text file.

Step 4 - Select row scope for import The purpose of this step is to identify which rows need to be imported. The rows to be imported must be contiguous in the file. The rules defined for importing rows from this file will then apply to all other files to be imported. Choose one of three options below.

The default option is to select all rows in the file. Alternatively, you can choose to take rows from a specific row number to a specific row number (use the preview window to identify row numbers) by entering the row numbers in the appropriate textboxes. Remember to press the enter key before proceeding. In addition, for situations where the data of interest lies between specific text markers, e.g., Begin Data and End Data, use option 3 to specify these markers; these markers must

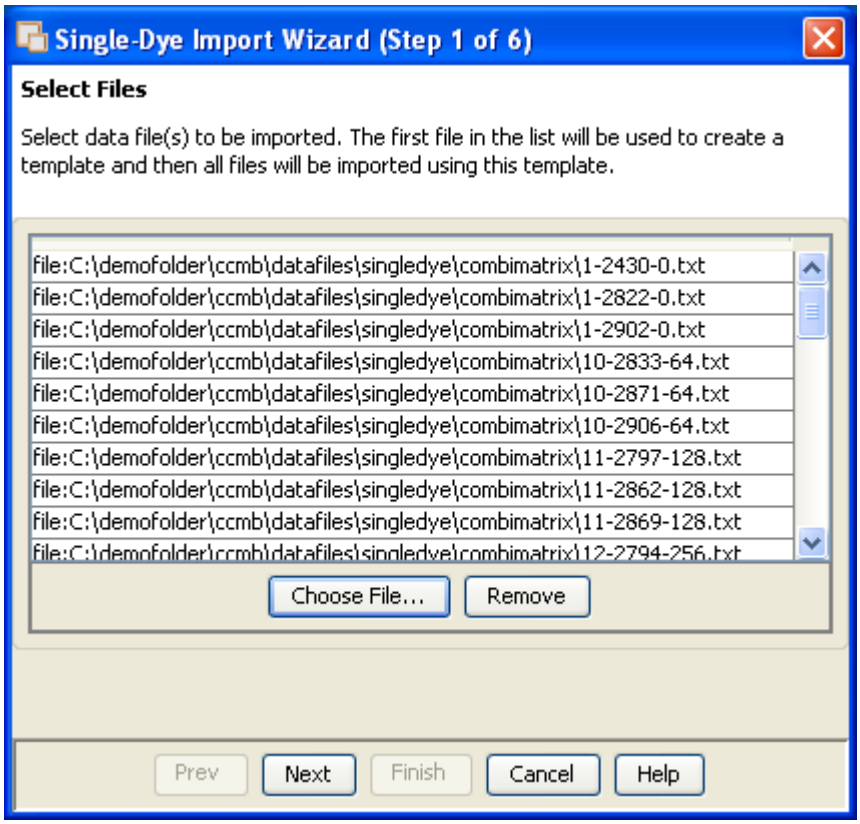


Figure 8.1: Step 1 of Import Wizard

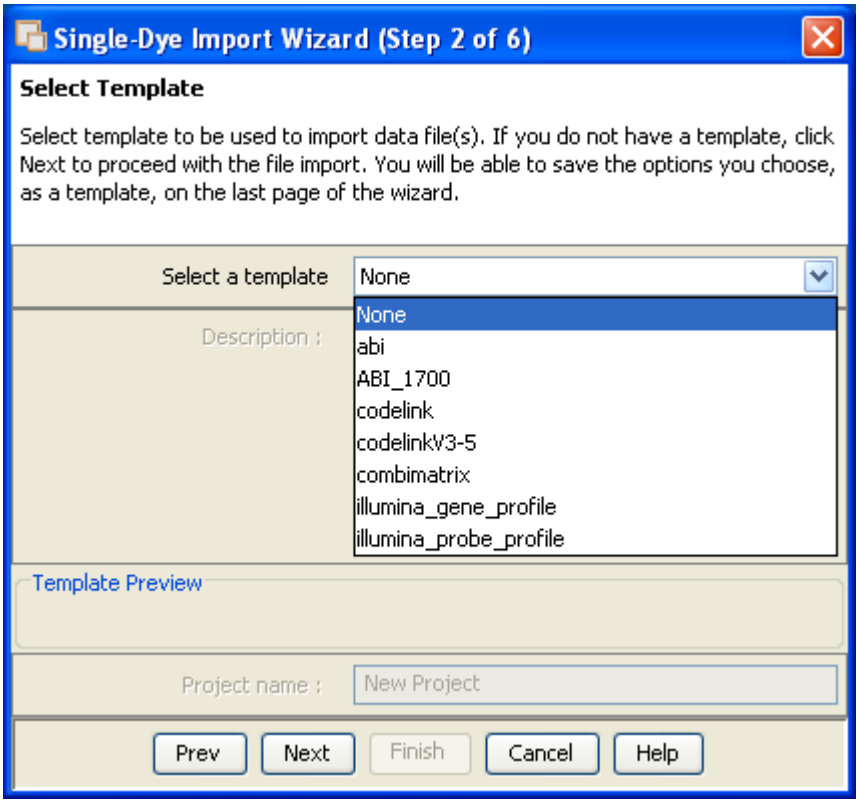


Figure 8.2: Step 2 of Import Wizard

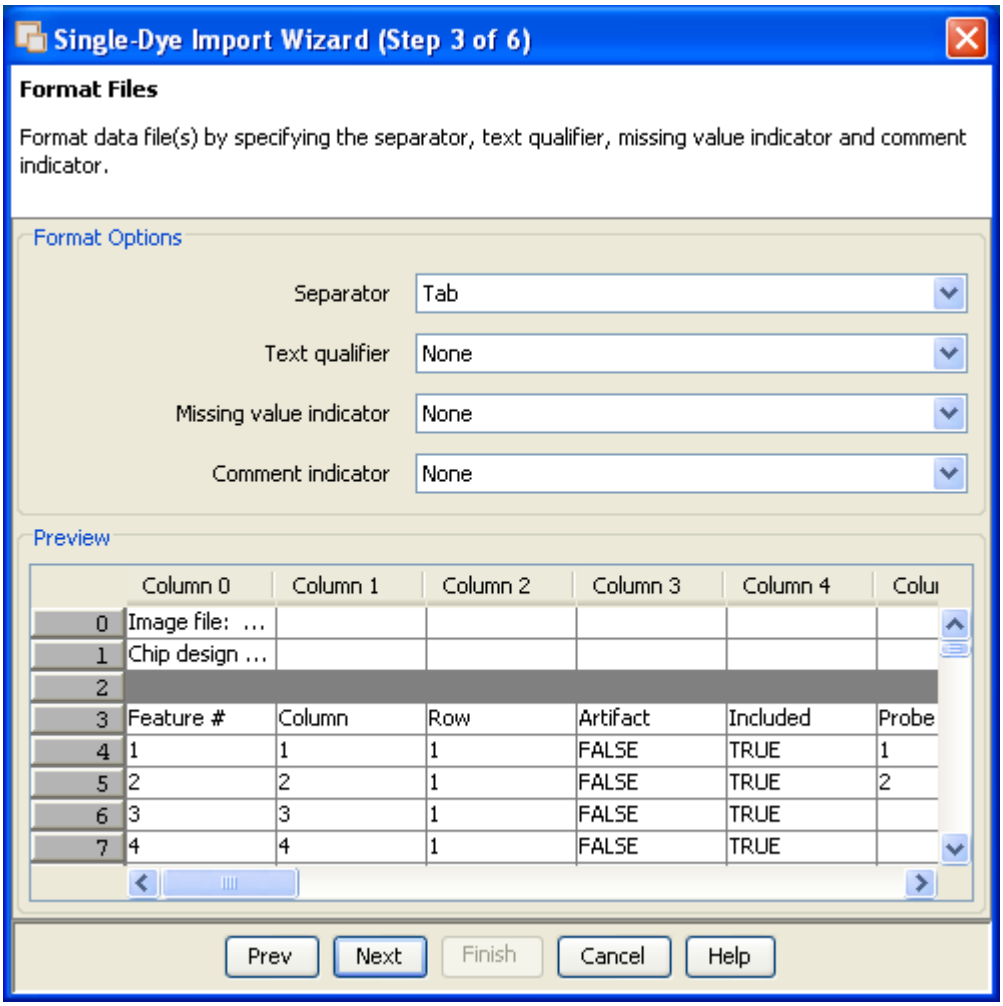


Figure 8.3: Step 3 of Import Wizard

appear at the very beginning of their respective lines and the actual data starts from the line after the first marker and ends on the line preceding the second marker. Note also that instead of choosing one of the options from the radio buttons, you can choose to select specific contiguous rows from the preview window itself by using Left-Click and Shift-Left-Click on the row header.

The panel at the bottom asks you to indicate whether or not there is a header row; in the latter case, dummy column names will be assigned.

Step 5 - Column Options and Column Marks The purpose of this step is to identify which columns are to be imported and what the type of each column is. The rules defined for importing rows from this file will then apply to all other files to be imported.

Select which columns need to be imported by checking/unchecking the textboxes on the left which appear against each column. In Column Options, specify how the columns selected by this procedure will be identified in other files to be imported; this identification can be done either by using the same *column names* or by using the same *column numbers*. The "column number" option is safer in instances where the actual column name could change from file to file, maybe due to addition of a date or the filename to the column name.

The Merge Options at the bottom specify how multiple files imported should be merged. Use the *alignment by row identifiers* option if the order of appearance of rows is not identical in all the files, and choose the *alignment by order of occurrence* otherwise. In the former case, you will need to mark one of the columns as an Identifier Column, as described below.

The most detailed task on this page is to provide a *Mark* for each column. The marks appear in the dropdown obtained by clicking on the None in the Column Mark panel against the relevant column. The set of available marks is listed below, with a brief explanation on what each mark means. Of these, only the Signals marks are compulsory.

Step 5 of the wizard requires identification of Column Marks. Marks along with Tags that are generated by **ArrayAssist** are used intelligently by the workflow browser to carry out the analysis. Tags and Marks are explained in detail below. The Column Mark column gives a drop down menu option to choose and match the data with the appropriate mark.

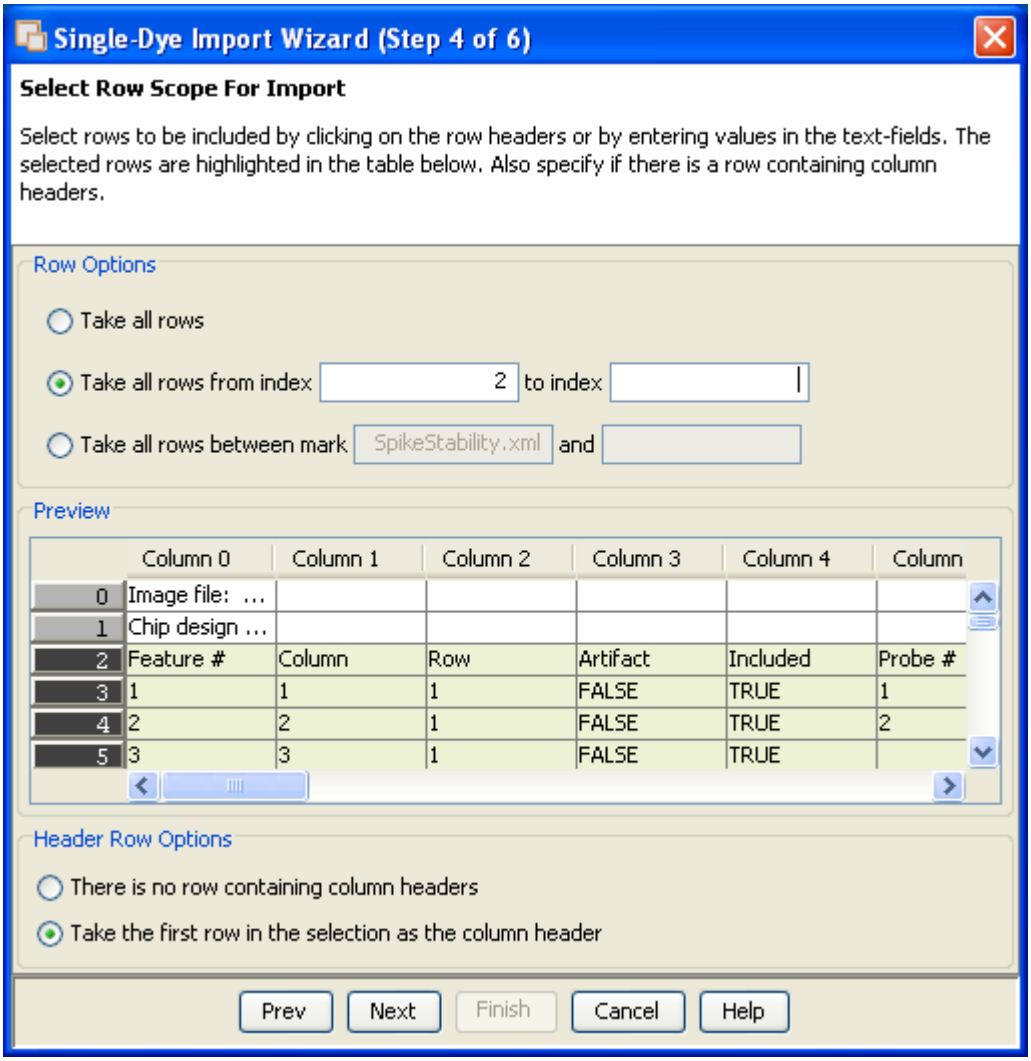

Single-Dye Import Wizard (Step 4 of 6)
Select Row Scope For Import
Select rows to be included by clicking on the row headers or by entering values in the text-fields. The selected rows are highlighted in the table below. Also specify if there is a row containing column headers.
Row Options
Take all rows
Take all rows from index 2 to index
Take all rows between mark SpikeStability.xml and
Preview
Column 0 Column 1 Column 2 Column 3 Column 4 Column
0 Image file: ...
1 Chip design ...
2 Feature # Column Row Artifact Included Probe #
3 1 1 1 FALSE TRUE 1
4 2 2 1 FALSE TRUE 2
5 3 3 1 FALSE TRUE
Header Row Options
There is no row containing column headers
Take the first row in the selection as the column header
Prev
Next
Finish
Cancel
Help

Figure 8.4: Step 4 of Import Wizard

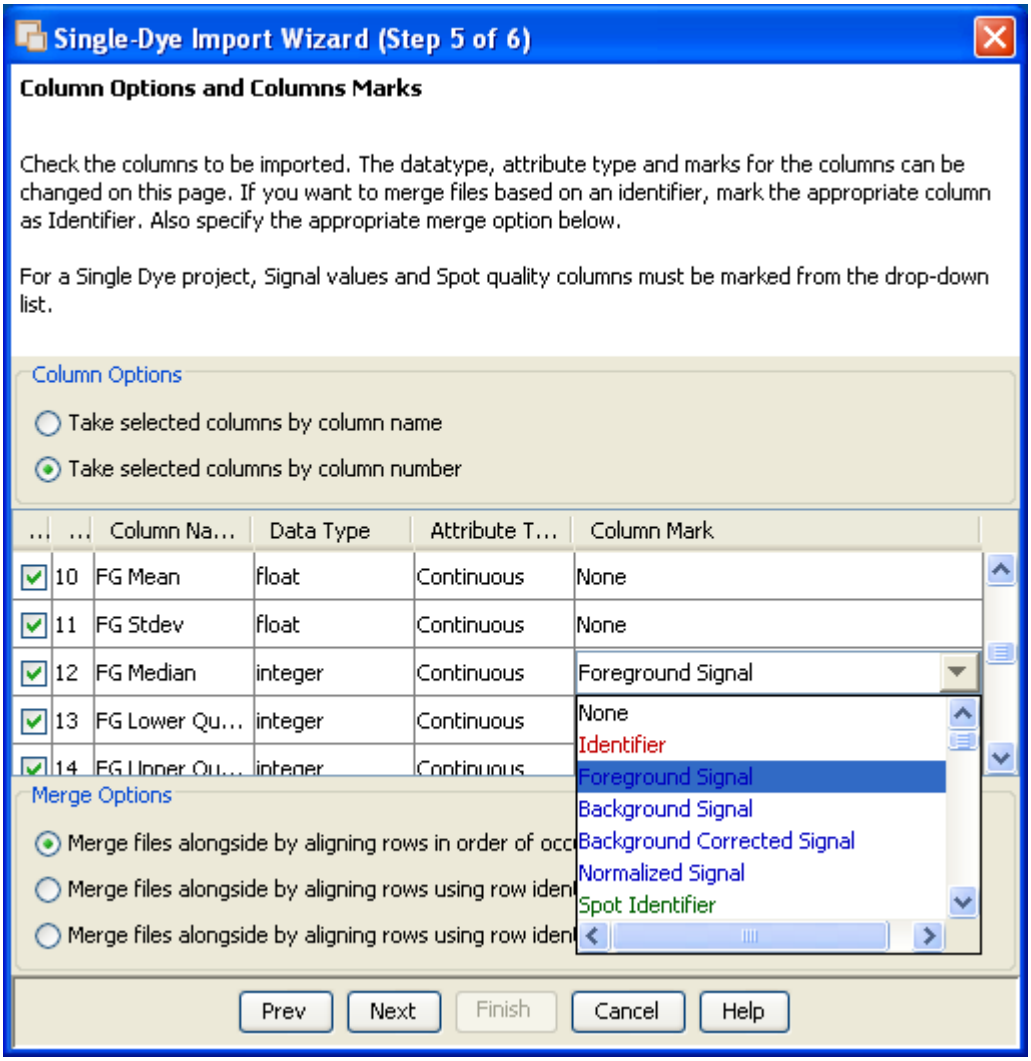

Single-Dye Import Wizard (Step 5 of 6)
Column Options and Columns Marks
Check the columns to be imported. The datatype, attribute type and marks for the columns can be changed on this page. If you want to merge files based on an identifier, mark the appropriate column as Identifier. Also specify the appropriate merge option below.
For a Single Dye project, Signal values and Spot quality columns must be marked from the drop-down list.
Column Options
Take selected columns by column name
Take selected columns by column number
... ... Column Na... Data Type Attribute T... Column Mark
10 FG Mean float Continuous None
11 FG Stdev float Continuous None
12 FG Median integer Continuous Foreground Signal
13 FG Lower Qu... integer Continuous
None
Identifier
Foreground Signal
Background Signal
Background Corrected Signal
Normalized Signal
Spot Identifier
Merge Options
Merge files alongside by aligning rows in order of occ
Merge files alongside by aligning rows using row iden
Merge files alongside by aligning rows using row iden
Prev
Next
Finish
Cancel
Help

Figure 8.5: Step 5 of Import Wizard

A Mark is associated with each spot property/data point being imported into the **ArrayAssist** spreadsheet. The broad categories of Marks are as follows:

- Signal Values
- The Spot Identifier and Coordinates Marks
- The Spot Type and Quality Marks
- Gene Annotation information

Associating data columns with Column Marks. This step asks for associating column names in the files with standard quantities associated with single-dye analysis. A list and explanation of these quantities appears below. Cretain columns are mandatory for a single-dye project, like the signal columns. For the remaining quantities, associating column marks is optional but may be useful for later steps, e.g., filtering, normalization etc. To associate a column with a quantity use the drop down menu.

Two warning notes are shown by **ArrayAssist** if there is no data associated with either Spot type or Flags. These messages are just for information. Flag is a quality parameter generated by the image analysis software. Spot type refers to specific controls like housekeeping genes, spike in genes, negative control genes etc.

- **Foreground intensity:** There could be multiple columns corresponding to the foreground intensity in the input files, e.g., mean foreground intensity or median foreground intensity; in such cases the median intensity is recommended over the mean intensity.
- **Background intensity:** There could be multiple columns, corresponding to the background intensity in the input files, e.g., mean background intensity or median background intensity; in such cases, the median intensity is recommended over the mean intensity. Typically, the same type of signal should be used for both background and foreground intensities. If foreground intensity is specified, the it is mandatory to mark the background intensity columns.
- **Background Corrected Intensity:** Some scanners will directly output background corrected intensities and call then the signal column. Normally, the file header my specify the background correction used. If these columns are available they should be marked as background corrected signal.

- **Normalized Background Corrected intensity:** Some scanners and output formats would output a normalized background corrected signal values. If these are present, such a column can be marked and will be brought into the dataset.
- **Identifier:** This is the row identifier in the dataset. If this is a unique column in the file, and identifies the gene or spot on the array, then the Identifier columns can be used to merge multiple files together. Certain scanner output formats or arrays may not output all the spots in the same order. Then the Identifier column must be used to merge multiple files or arrays and brought into **ArrayAssist** by explicitly chosing the option to merge files alongdside by aligning rows using the row Identifiers in the merge option at the bottom of the page.
- **Spot Identifier:** This is an optional field. Each spot typically has a spot number on the chip. If the spot identifier is used to merge rows, then this column must be marked as an Identifier column.
- **Physical X and Y Spot Coordinates:** These are optional and are required to view a physical image of the chip via scatter plots in **ArrayAssist**.
- **Block Number(s):** Typically, spotted arrays are spotted in blocks. These blocks are numbered either with block-row and block-column numbers or with single numbers from 1 to the number of blocks; select one of these two options. This field is optional but useful if you want to normalize data in each block separately.
- **Flags:** Each spot has an associated flag which can be turned on in the image analysis step to indicate that the spot is bad. These flags will be useful for filtering spots.
- **Spot p-value:** Some Image analysis software output a p-value based on the error model used in the computation of each log ratio.
- **Gene Description:** The purpose of this is purely to carry over gene description information to the output dataset.
- **Other Annotation Marks:** If the dataset contains other anotation columns like the GeneBank Accession Numner, the Gene Name, etc, these columns can be marked on the dataset while importing data into **ArrayAssist**. If the dataset contains such

annotation columns, they can used for running the annotation workflow or launching the genome browser.

- **Duplicate and New Marks.** Other than signals, **ArrayAssist** will not allow the same mark to be used for multiple columns. New marks can be defined by choosing the EnterNew towards the bottom of the marks dropdown list; however, filtering based on newly defined marks will not be possible via the current workflow steps and will need to be performed manually, i.e., using the filter utility or by writing a script etc.

Tags are associated with various forms of raw data and comprise of the following. Depending upon the the columns that are marked in the input files, datasets corresponding to the vaious tages will be automatically created in the project.

- Raw Signals - Foreground and Background
- Background corrected signal
- Normalized background corrected signal values

NOTE: All panels and the whole window is resizable by dragging if needed. Also if Spot Type or Flag is not marked then a warning is issued before proceeding.

Step 6 - Summary This step shows a summary of all the options chosen for building the template. Use the *Template name* to provide a name for this template. The template will be saved and can be subsequently used to import other files that have the same format. Use the *Project name* option to provide a name for the project being created.

This is the last step in the wizard, choose *Finish* to bring the data into **ArrayAssist** for further analysis using the Workflow Browser.

Once the single-dye data is loaded into **ArrayAssist**, a normal analysis flow can be performed by the use of the workflow browser. The steps in the workflow browser captures the most common two-dye analysis workflow.

NOTE: If the import wizard returns with an error, then there is a mismatch between the template used and the files input. Please send mail to `techservices@stratagene.com`with a description of the error message along with one or two sample files.

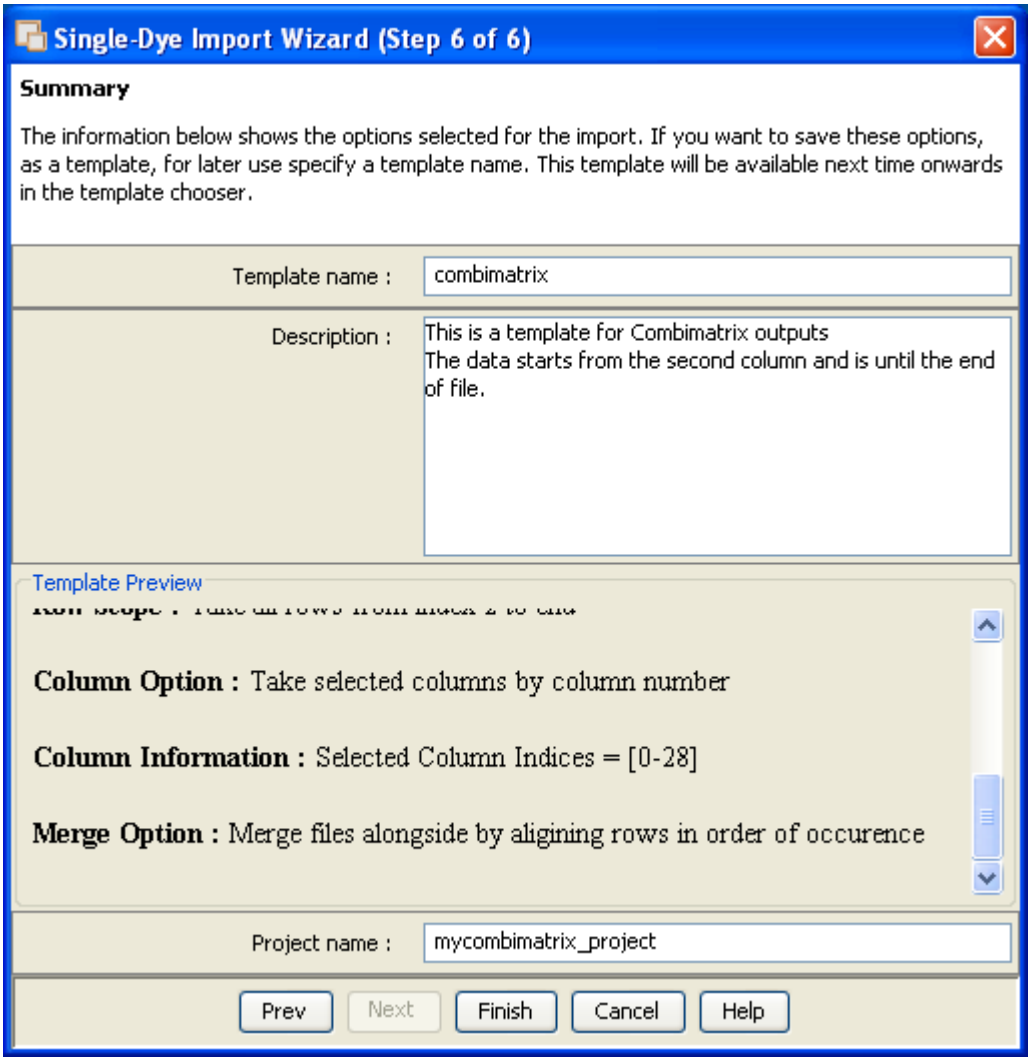

Single-Dye Import Wizard (Step 6 of 6)
Summary
The information below shows the options selected for the import. If you want to save these options, as a template, for later use specify a template name. This template will be available next time onwards in the template chooser.
Template name :
combimatrix
Description :
This is a template for Combimatrix outputs
The data starts from the second column and is until the end of file.
Template Preview
Column Option : Take selected columns by column number
Column Information : Selected Column Indices = [0-28]
Merge Option : Merge files alongside by aligining rows in order of occurence
Project name :
mycombimatrix_project
Prev
Next
Finish
Cancel
Help

Figure 8.6: Step 6 of Import Wizard

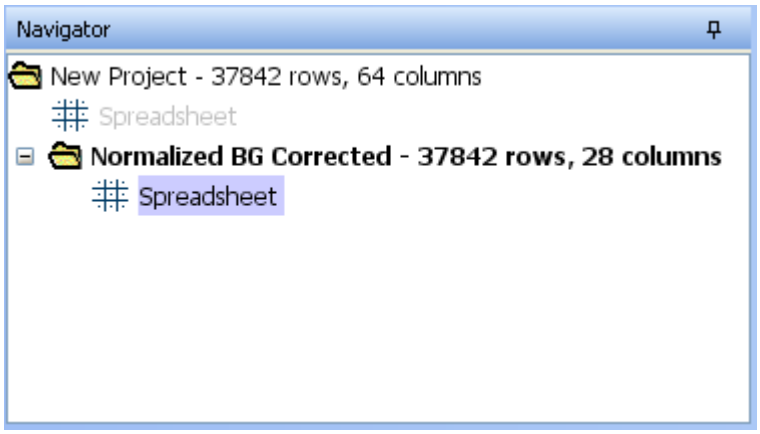


Figure 8.7: The Navigator at the Start of the Single Dye Workflow

8.2 The Single-Dye Analysis Workflow

After creating the appropriate template, use *File* $\longrightarrow$*Import SingleDye* wizard to import files using this template. Select the files of interest and select the template from the drop-down list of all templates. Successful import now will result in the creation of a new single-dye project. The navigator on the left should show the number of rows in the project (which corresponds to the number of probes on one array) and the number of columns (which includes all type of signals, flags and ids).

The Initial Datasets. In addition, the navigator should show either a Raw dataset, a BG (background) Corrected dataset, or a Normalized BG Corrected dataset. More than one of these datasets could also be shown depending upon which type of signals were marked in the template creation process. If Foreground and Background Signals were marked then a raw dataset containing foreground and background values for each array imported will be shown, and likewise, for Background Corrected and Normalized signal values. In addition to the signal columns, all these datasets will contain all other columns marked in the template creation process. The list of columns and their types and marks can be seen using Data Properties icon. If you used a template that came prepackaged with **ArrayAssist**, then you may not be familiar with the notion of *column marks*; refer to Section Column Options and Marks for details.

NOTE: If the navigator does not show any of Raw, BG Corrected or Normalized, then the template used for import did not have signals marked correctly. Go back and create a new template making sure that signal columns are marked appropriately this time or send emailx to techservices@stratagene.comto request support.

NOTE: Most datasets and views in **ArrayAssist** are *lassoed*, i.e., selecting one or more rows/columns/points will highlight the corresponding rows/columns/points in all other datasets and views. In addition, if you select probes from any dataset or view, signal values and gene annotations for the selected probes can be viewed using *View* $\longrightarrow$*Lasso* (you may need to customize the columns visible on the Lasso view using Right-Click *Properties*).

The Workflow. Once the project opens up with the appropriate datasets in the navigator then the primary analysis steps are enumerated in the workflow browser panel on the right. These steps can be run by clicking upon the corresponding links. A listing and explanation of these steps appears in the sections below.

NOTE: Steps in the workflow browser are related to the dataset that is in focus in the navigator. Each step operates on the dataset in focus. Further, it may or may not be applicable to this dataset. Before running a specific step, you may need to move focus to the relevant dataset in the navigator.

8.2.1 Getting Started

Click on this link to take you to the chapter on Analyzing Single-Dye Data.

8.2.2 The Experiment Grouping

The very first step is providing Experiment Grouping. The *Experiment Grouping* view which comes up will initially just have the imported file names. The task of grouping will involve providing more columns to this view containing *Experiment Factor* and *Experiment Grouping* information. A Control vs. Treatment type experiment will have a single factor comprising 2 groups, Control and Treatment. A more complicated Two-Way experiment could feature two experiment factors, genotype and dosage, with genotype having transgenic and non-transgenic groups, and dosage having



Figure 8.8: The Single Dye Workflow Browser

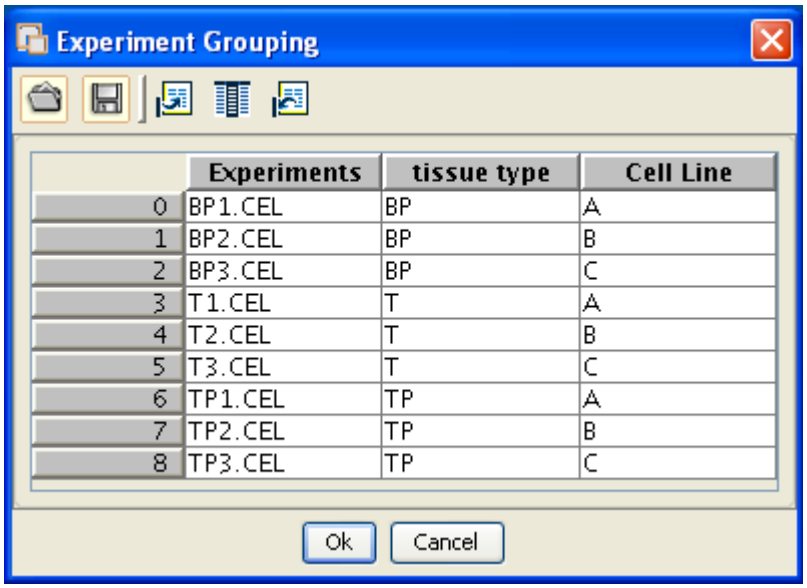


Figure 8.9: The Experiment Grouping View With Two Factors

5, 10, and 50mg groups. Adding, removing and editing Experiment Factors and associated groups can be performed using the icons described below.

Reading Factor and Grouping Information from Files. Click on the Read Factors, Groups from File icon to read in all the Experiment Factor and Grouping information from a tab or comma separated text file. The file should contain a column containing imported file names; in addition, it should have one column per factor containing the grouping information for that factor. Here is an example tab separated file. The result of reading this tab file in is the new columns corresponding to each factor in the Experiment Grouping view.

```
#comments
#comments
filename genotype dosage
A1.GPR    NT       0
A2.GPR     T       0
A3.GPR    NT      20
A4.GPR     T      20
A5.GPR    NT      50
```

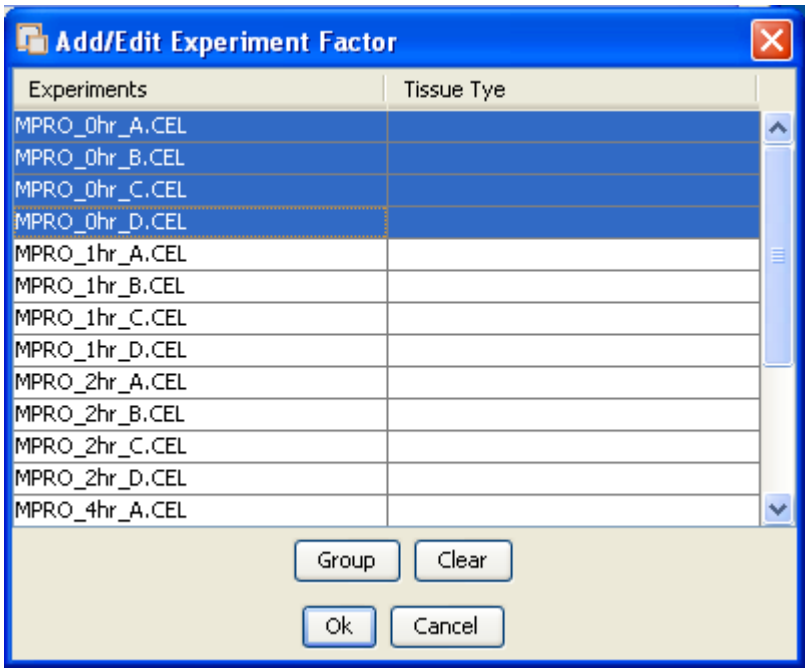


Figure 8.10: Specify Groups within an Experiment Factor

```
A6.GPR      T       50
```

Adding a New Experiment Factor. Click on the Add Experiment Factor icon to create a new Experiment Factor and give it a name when prompted. This will show the following view asking for grouping information corresponding to the experiment factor at hand. The files shown in this view need to be grouped, with each group comprising biological replicate arrays. To do this grouping, select a set of imported files, then click on the *Group* button, and provide a name for the group. Selecting files uses Left-Click , Ctrl-Left-Click , and Shift-Left-Click , as before.

Editing an Experiment Factor. Click on the Edit Experiment Factor icon to edit an Experiment Factor. This will pull up the same grouping interface described in the previous paragraph. The groups already set here can be changed on this page.

Remove an Experiment Factor. Click on the Remove Experiment Factor icon to remove an Experiment Factor.

8.2.3 Primary Analysis

This section includes links to do primary analysis of single-dye data. They include methods to supress bad spots in the data, vaious methods of background correction, normalization, quality assessment and data transformations. These are detailed below:

Suppressing Bad Spots This is a quality control step and is optional. This link can be used to filter based on flags generated by the image analysis software or based on the signal values. Typically, low signal values are filtered to remove noise from the data. The pop up window has two tabs, one for filtering on flags and the other for filtering on signals.

This step will create a new dataset in which signal values corresponding to bad spots are replaced by missing values; all further operations can be performed on this dataset. Bad spots can be identified by quality marks The Spot Type and Quality Marks or by signal value ranges. The signal value used is the one present in the dataset that is in focus in the navigator.

Background Correction Background Correction is admissible only on the Raw dataset containing Foreground and Background signal values. Correction is usually performed by subtracting the background value for a spot from its foreground value (the FG-BG option) or alternatively, subtracting an averaged chip background value from the foreground value for each spot (the FG- Mean/Median BG option). Further, **ArrayAssist** offers background correction by subtracting an average of the Negative Control spots on the chip, where the negative control spots are indicated by the Spot Type mark The Spot Type and Quality Marks. Finally, **ArrayAssist** also offers a way to subtract a fixed constant from all FG values using the FG - constant option.

There are four choices for background correction

- Foreground - constant: This option can be used to subtract a constant value from all the foreground intensities. Select zero (0) if no correction needs to be done.
- FG-BG: This option is used to subtract background intensities from their respective foreground intensities.
- FG-Mean/Median of BG: This option is used to subtract either the mean or the median of the background from all foreground intensities for each channel on all arrays.

- FG-Mean/Median of Negative Control spots: This option is used to subtract either the mean or median of negative control spots from all foreground intensities for each channel on all arrays.

NOTE: If you did not mark any column as Spot Type while creating the template or if you wish to create and mark a new column containing negative control indicators as Spot Type, then select the probes of interest on the spreadsheet, use *Data* $\longrightarrow$*Row Operations* $\longrightarrow$*Label Rows* to label the negative control probes, then use **Data** $\longrightarrow$**Properties** to mark this newly added Label column as the Spot Type column.

NOTE: Background Correction could result in negative values, which could create problems later. You can suppress negative values using the Suppress Bad Spots link in the workflow browser; suppress spots where the background corrected signal is less than 0.

Normalization The next step in the analysis is normalization.

Normalization is admissible only on Background Corrected datasets. If for some reason you do not wish to perform background correction but wish to go on to normalization directly, then use the FG-constant background correction method with the constant set to 0 to derive a background corrected dataset.

- Mean/Median scale: The most common normalization method is to equalize the array means or medians by scaling (Mean/Median Scale Option); you will need to provide the target value which all medians/means attain after normalization.
- Mean/Median scale using Housekeeping genes: The Mean/Median scaling using Housekeeping genes option is useful in situations where most genes on the chip are changing is response to stimulus and therefore equalizing means/medians does not make sense. In this situation, the means/medians of housekeeping spots are equalized across chips by scaling. Housekeeping spots are identified using the Spot Type mark (as was the case for negative controls in background correction Background Correction).
- Lowess Against baseline: The Lowess option is useful when there are non-linear non-biological distortions across arrays. To run

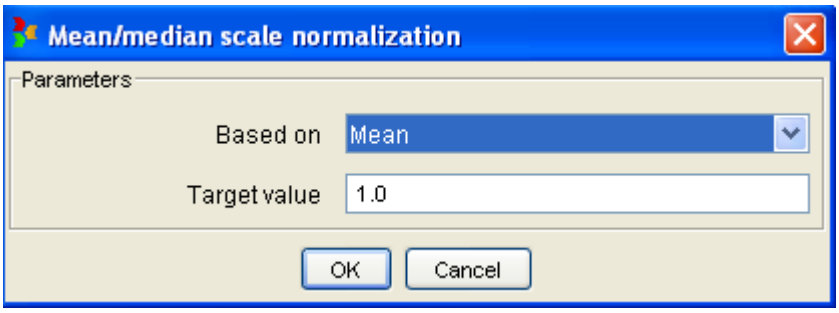


Figure 8.11: Normalization

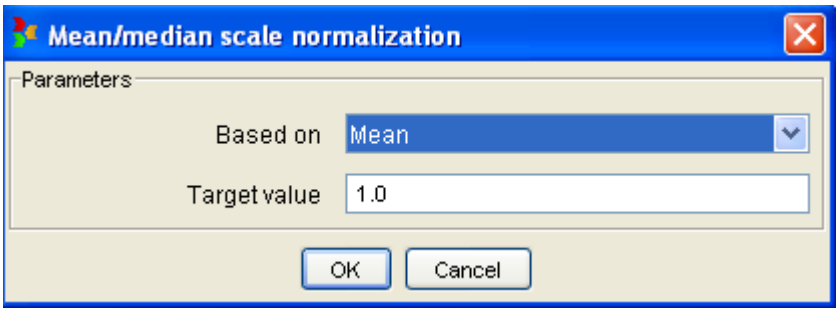


Figure 8.12: Normalization

Lowess, you will need to denote one of the experimental groups identified (The Experiment Grouping) as the baseline group; the average of all arrays in the baseline group is used as the baseline array for Lowess normalization.

The advantage of Lowess over MeanShift is that Lowess is a more powerful method because of its ability to perform differential correction in different intensity ranges while MeanShift is much coarser; it uses the same correction everywhere.

Quality Assessment The quality assessment step has a few visualization options to check the quality of the data. This step can be used to decide the data points to carry forward for further analysis.

- **Data Quality Plots** This step is for checking visual consistency across arrays, i.e., whether the data is well normalized or not. Clicking on this link will output a scatter plot, a matrix plot, and a statistics view. The scatter plot will show the first two arrays; other arrays can be viewed by changing the X and Y axes using the drop-down list. The matrix plot will show by default the first 3 arrays. More arrays can be viewed using Right-Click *Properties*⟶*Rendering*⟶*Page*, and changing the numbers of rows and columns (*remember to press enter after putting in each value.*). These two plots should produce approximately 45 degree plots for the arrays to be consistent. Sometime the scatter plots are better viewed on the log scale, which can be set via Right-Click *Properties*. The statistics plot shows distributions of signal values within each array, which should also be consistent across arrays.
- **Principal Component Analysis on Arrays.** This link will perform principal component analysis on the arrays. It will show the standard PCA plots (see PCA for more details). The most relevant of these plots used to check data quality is the *PCA scores* plot, which shows one point per array and is colored by the Experiment Factors provided earlier in the *Experiment Grouping* view. This allows viewing of separations between groups of replicates. Ideally, replicates within a group should cluster together and separately from arrays in other groups. The *PCA scores* plot can be color customized via Right-Click *Properties*. All the Experiment Factors should occur here, along with the Principal Components E0, E1 etc. The PCA Scores view is *lassoed*, i.e.,

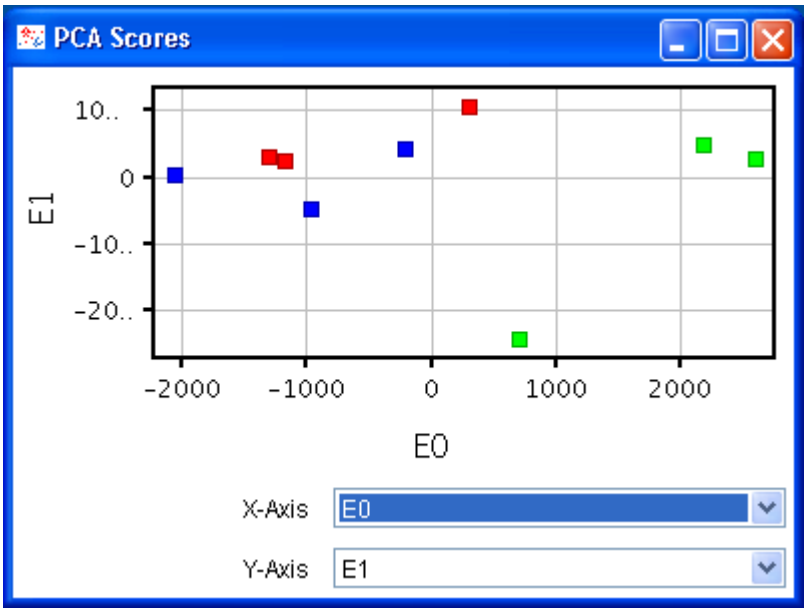


Figure 8.13: PCA Scores Showing Replicate Groups Separated

selecting one or more points on this plot will highlight the corresponding columns (i.e., arrays) in all the datasets and views. Further details on running PCA appear in Section PCA.

- **Correlation Plots.** This link will perform correlation analysis across arrays. It finds the correlation coefficient for each pair of arrays and then displays these in two forms, one in textual form as a correlation table view, and other in visual form as a heatmap. The heatmap is colorable by Experiment Factor information via Right-Click *Properties*. The intensity levels in the heatmap can also be customized here. The text view itself can be exported via Right-Click *Export as Text*. Note that unlike most views in **ArrayAssist**, the correlation views are not *lassoed*, i.e., selecting one or more rows/columns here will not highlight the corresponding rows/columns in all the other datasets and views. Sometimes it is useful to cluster the arrays based on correlation. To do this, export the correlation text view as text, then open it via *File*$\longrightarrow$*Open*, and then use *Cluster*$\longrightarrow$*Hier* to cluster. Row labels on the resulting dendrogram can then be colored based on Experiment Factors using Right-Click *Properties*.

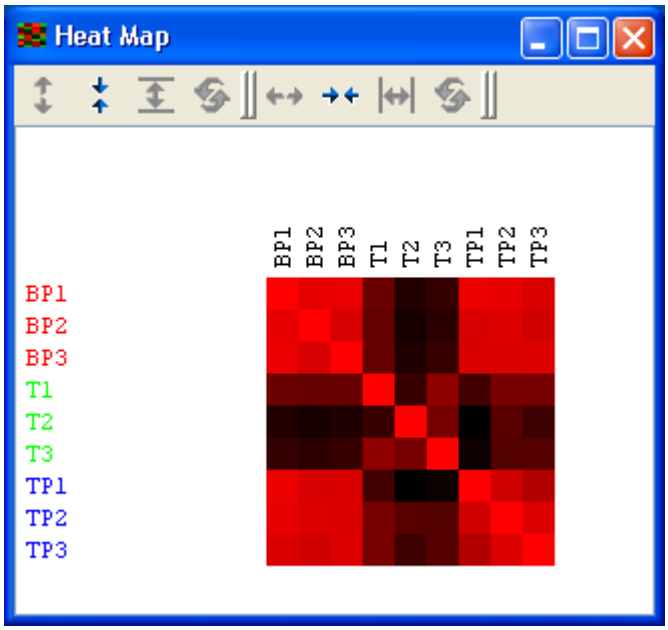
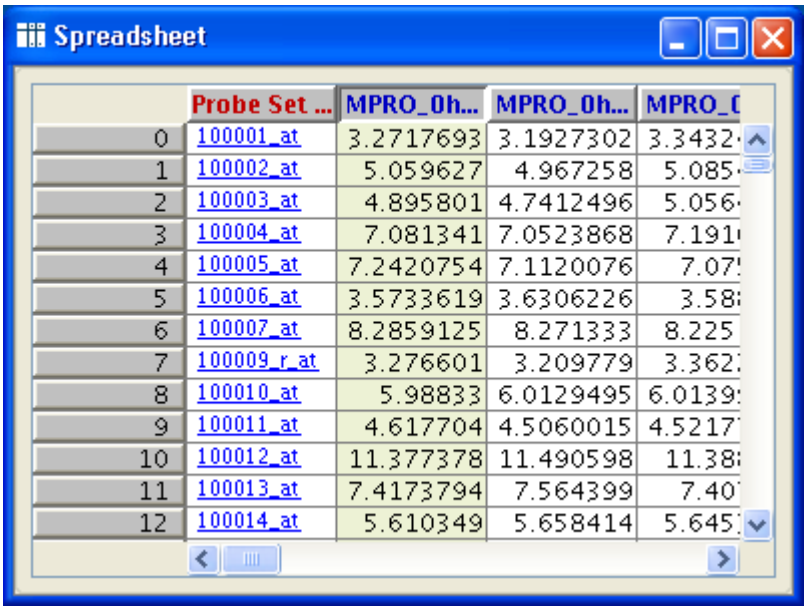


Figure 8.14: Correlation HeatMap Showing Replicate Groups Separated

Data Transformations Once data quality has been checked for, the next step is to perform various transformations. The list of transformations available in the workflow browser is described below. Each transformation will produce a new child dataset in the navigator. Also, rows and columns in each of these datasets will be *lassoed* with the rows and columns, respectively, in all the other datasets. Selecting a row/column in one dataset with highlight it in all the other datasets and open views, making it easy to track objects across datasets and views.

NOTE: Data transformation will often require you to select a specific dataset in the navigator. For example, Log-Transformation will require selecting a Summarization dataset containing signal values (obtained via one of the summarization algorithms or via the import of CHP files). Appropriate messages will be displayed if the right dataset is not selected in the Navigator.

- **Variance Stabilization.** Use this step to add a fixed quantity (16 or 32) to all linear scale signal values. This is often performed



	Probe Set ...	MPRO_0h...	MPRO_0h...	MPRO_(
0	100001_at	3.2717693	3.1927302	3.3432
1	100002_at	5.059627	4.967258	5.085
2	100003_at	4.895801	4.7412496	5.056
3	100004_at	7.081341	7.0523868	7.191
4	100005_at	7.2420754	7.1120076	7.07
5	100006_at	3.5733619	3.6306226	3.58
6	100007_at	8.2859125	8.271333	8.225
7	100009_r_at	3.276601	3.209779	3.362
8	100010_at	5.98833	6.0129495	6.0139
9	100011_at	4.617704	4.5060015	4.5217
10	100012_at	11.377378	11.490598	11.38
11	100013_at	7.4173794	7.564399	7.40
12	100014_at	5.610349	5.658414	5.645

Figure 8.15: New Child Dataset Obtained by Log-Transformation

to suppress noise at log signal values, e.g., as shown in the pre- and post- variance stabilization scatter plots generated by PLIER summarization. Log transformation should be performed only after variance stabilization.

- **Log Transformation.** Use this step to convert linear scale data to logscale, where logs are taken to base 2. This step is *necessary* before performing statistics, baseline transformations and computing sample averages; these transformations will work only on log-transformed summarized datasets.

- **Baseline Transformation.** This step only works on log-transformed datasets and produces *log-ratios* from log-scale signals. The ratios are taken relative to the average value in a specified experiment group called the *Baseline* group.

  Recall that experiment factors and groups were provided earlier as in Section 5.3.2. One of these groups of replicate arrays will serve as the baseline. Next, the log-scale signal values of each probeset will be averaged over all arrays in the baseline group. This amount will be subtracted from each log-scale sig-

nal value for this probeset in the log-transformed summarized dataset. This transform is useful primarily for viewing (e.g., in a heatmap, colors in the baseline group are subdued and all others reflect a color relative to this baseline group, in particular, positive and negative log ratios relative to this group are well differentiated).

To run this transformation, you will need to specify the baseline group. To this effect, **ArrayAssist** will ask you first to choose an experiment factor amongst those provided prior to generating signal values. Next, it will ask you to choose the baseline group from within the groups for this experiment factor.

- **Compute Sample Averages.** This step only works on log-transformed datasets and averages arrays within the same replicate groups to obtain a new set of averaged arrays. Recall that experiment factors and groups were provided earlier as in Section on The Experiment Grouping. To run this transformation, you will need to specify the experiment factor(s) and group(s) over which averaging needs to be performed. For instance, you may choose one experiment factor and all or a few groups corresponding to this factor; the averages within each of the chosen groups will be computed. If you choose multiple experiment factors, say factor A with groups AX and AY and factor B with groups BX and BY, then averages will be computed within the 4 groups, AX/BX, AX/BY, AY/BX, and AY/BY. The result of running this transformation will be a new dataset containing the group averages. By using the up/down arrow keys on the dialog shown below, the order of groups in the output dataset can be customized.
- **Fill In Missing Values.** This step only works on log-transformed datasets and allows missing values in signal columns to be filled in either by a fixed value or via interpolation using the KNN (K Nearest Neighbours) algorithm.
  - Fixed value: All missing values will be replaced by a fixed value. The choice of the fixed value can be entered in the pop up window in 'Replace by' field.
  - KNN Algorithm: The KNN algorithm can be used to fill in all missing values.

  The second tab in the pop up window called *Columns* can be used to pick columns for filling in missing values.

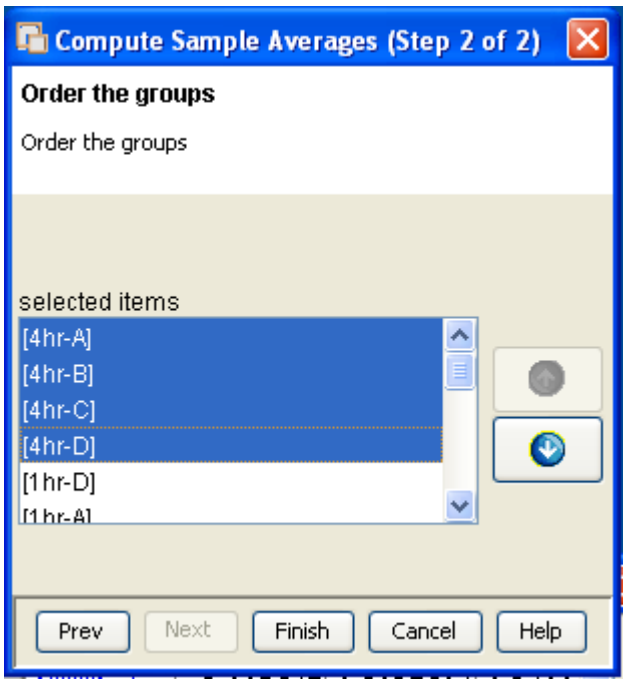


Figure 8.16: Reorder Groups for Viewing

- **Combine Replicate Spots.** This step averages over replicate spots on the arrays. Replicates are identified based on values in a specified column. Note that the averaging works in place, i.e., the average value is repeated for each of the replicate spots rather than reducing each group of replicate spots to one spot each.

8.2.4 Data Viewing

Data in datasets within an Single Dye project can be visualized via the views in the *Views* menu as well as the view icons on the toolbar. Each view allows various customizations via the Right-Click *Properties* menu. Some views which operate on specific columns or subsets of columns will use the column selection in the currently active dataset by default. To select columns in a dataset use Left-Click , Ctrl-Left-Click , Shift-Left-Click on the body of the column (and not on the header). For more details on the various views and their properties, see Data Visualization.

The Single Dye Workflow browser currently provides the following additional viewing options.

Profile Plot by Group This view option allows viewing of profiles of probesets across arrays comprising specific experiment factors and groups of interest. Recall that experiment factors and groups were provided earlier as in Section The Experiment Grouping. To obtain this plot, you will need to specify the experiment factor(s) and group(s) over which averaging needs to be performed. For instance, you may choose one experiment factor and all or a few groups corresponding to this factor; you can then also use the up/down arrows to specify the order in which the various groups will appear on the plot. A profile plot with the arrays comprising these groups, in the right order, will be presented.

8.2.5 Significance Analysis

ArrayAssist provides a battery of statistical tests including t-tests, Mann-Whitney Tests, Multi-Way ANOVAs and One-Way Repeated Measures tests. Clicking on the *Significance Analysis Wizard* will launch the full wizard which will guide you through the various testing choices. Details of these choices appear in The Differential Expression Analysis Wizard, along with detailed usage descriptions. For convenience, a few commonly used tests are encapsulated in the Single-Dye Workflow as single click links; these are described below.

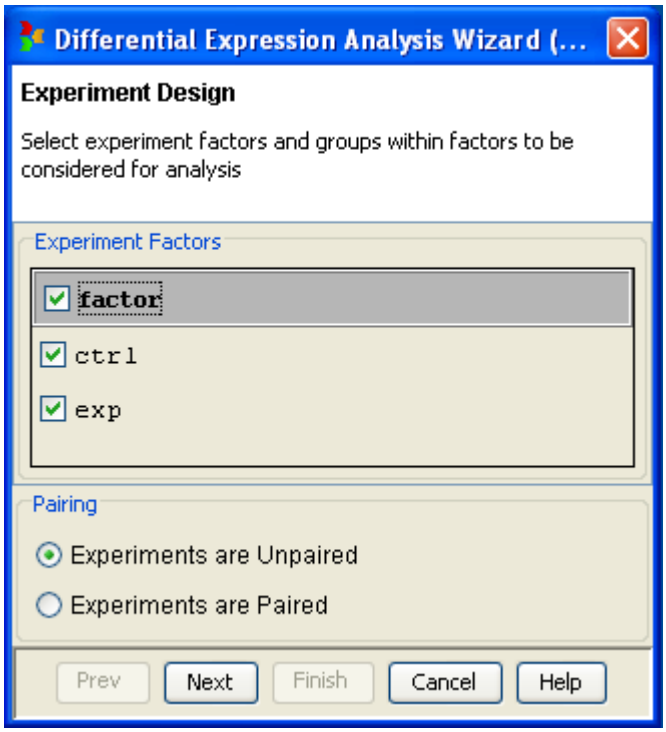


Figure 8.17: Significance Analysis Steps in the Singledye Analysis Workflow

NOTE: Significance Analysis requires that Factor and Group information be provided BEFORE signal values are generated. Also the single-click links can only be performed on log-transformed datasets.

The Treatment vs Control t-test: This link will function only if the Experiment Grouping view has only one factor, which comprises two groups. You will be prompted for which of the two groups is to be considered as the Control group. A standard t-test is then performed between Treatment and Control groups. p-values, Fold Changes, Directions of Regulation (up/down), and Group Averages are derived for each probeset in this process. In addition, p-values corrected for multiple testing are also derived using the Benjamini-Hochberg FDR method (see Differential Expression Analysis for details).

The Multiple Treatments vs Control t-test: This link will function only

if the Experiment Grouping view has only one factor, which comprises more than two groups. You will be prompted for which of the groups is to be considered as the Control group. Subsequently, each non-Control group will be t-tested against the Control group. p-values, Fold Changes, Directions of Regulation (up/down), and Group Averages are derived for each probeset in each t-test. In addition, p-values corrected for multiple testing are also derived using the Benjamini-Hochberg FDR method (see Differential Expression Analysis for details).

Multiple Treatments ANOVA: This link will function only if the Experiment Grouping view has only one factor, which comprises more than two groups. A One-Way ANOVA will be performed on all these groups. p-values and Group Averages are derived for each probeset in this process. In addition, p-values corrected for multiple testing are also derived using the Benjamini-Hochberg FDR method (see Differential Expression Analysis for details).

Significance Analysis Wizard This link invokes the differential expression wizard. This can be used to run any parametric or non-parametric statistical test along with options for multiple testing correction. Use this option if the experiment set up does not fall into one of the above categories.

Results of Significance Analysis are presented in views and datasets described below. All of these appear under the *Diffex* node in the navigator as shown below.

The Statistics Output Dataset. This dataset contains the p-values and fold-changes (and other auxiliary information), generated by Significance Analysis.

The Differential Expression Analysis Report. This report shows the test type and the method used for multiple testing correction of p-values. In addition, it shows the distribution of genes across p-values and fold-changes in a tabular form. For t-tests, each table cell shows the number of genes which satisfy the corresponding p-value and fold-change cutoffs. For ANOVAs, each table cell shows the number of genes which satisfy the corresponding fold-change cutoff only. For multiple t-tests, the report view will present a drop down box which can be used to pick the appropriate t-test. Clicking on a cell in these tables will select and lasso the corresponding genes in all the views. Finally, note that the last row in the table shows some

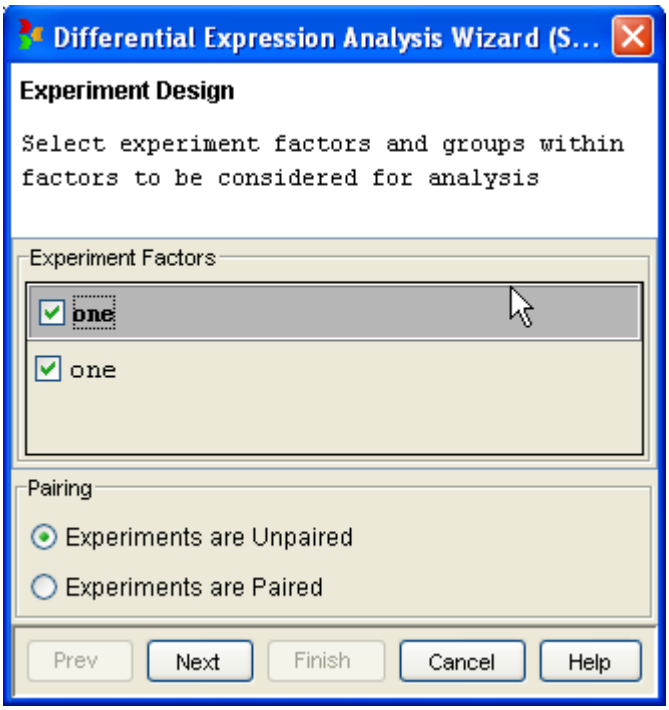


Figure 8.18: Step 1 of Differential Expression Analysis

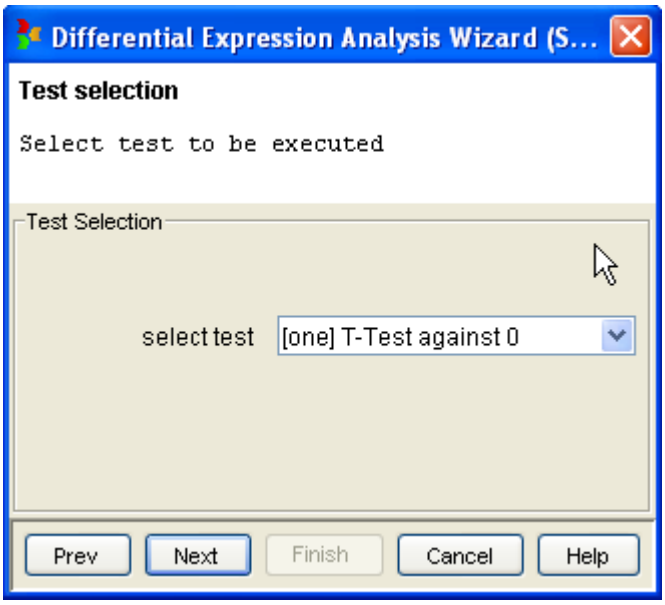


Figure 8.19: Step 2 of Differential Expression Analysis

Expected by Chance numbers. These are the number of genes expected by pure chance at each p-value cut-off. The aim of this feature is to aid in setting the right p-value cutoff. This cut-off should be chosen so that the number of gene expected by chance is much lower then the actual number of genes found (see Differential Expression Analysis for details).

The Volcano Plot. This plot shows the log of p-value scatter-plotted against the log of fold-change. Probesets with large fold-change and low p-value are easily identifiable on this view. The properties of this view can be customized using Right-Click *Properties*.

Filtering on p-values and Fold Changes. Finally, once significance analysis has been done, the dataset can be filtered to extract genes that are significantly expressed. Click on the link and this will pop-up a dialog to provide the significance value and the fold change criteria. This will create a child dataset with the set of genes that satisfy the filter critera provided.

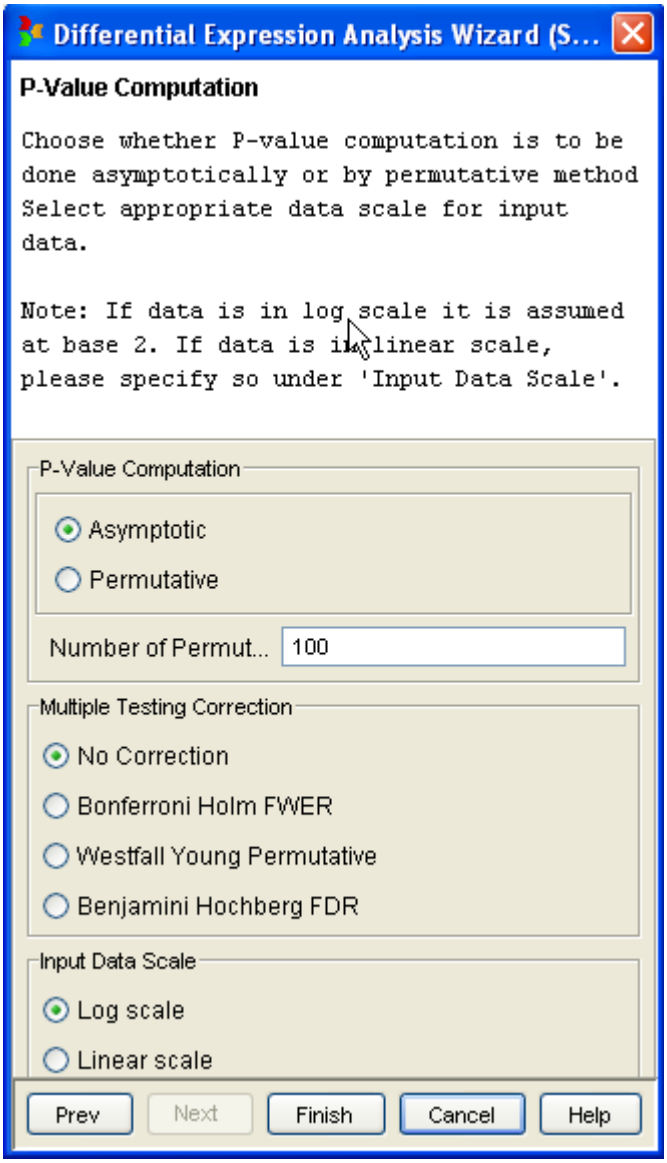

Differential Expression Analysis Wizard (S...
P-Value Computation
Choose whether P-value computation is to be done asymptotically or by permutative method Select appropriate data scale for input data.
Note: If data is in log scale it is assumed at base 2. If data is in linear scale, please specify so under 'Input Data Scale'.
P-Value Computation
Asymptotic
Permutative
Number of Permut...
100
Multiple Testing Correction
No Correction
Bonferroni Holm FWER
Westfall Young Permutative
Benjamini Hochberg FDR
Input Data Scale
Log scale
Linear scale
Prev
Next
Finish
Cancel
Help

Figure 8.20: Step 3 of Differential Expression Analysis

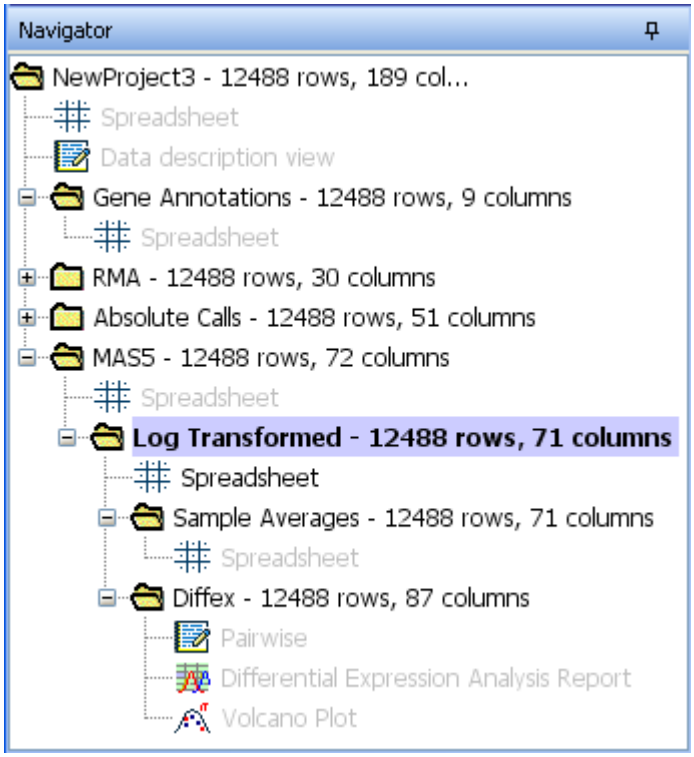


Figure 8.21: Navigator Snapshot Showing Significance Analysis Views

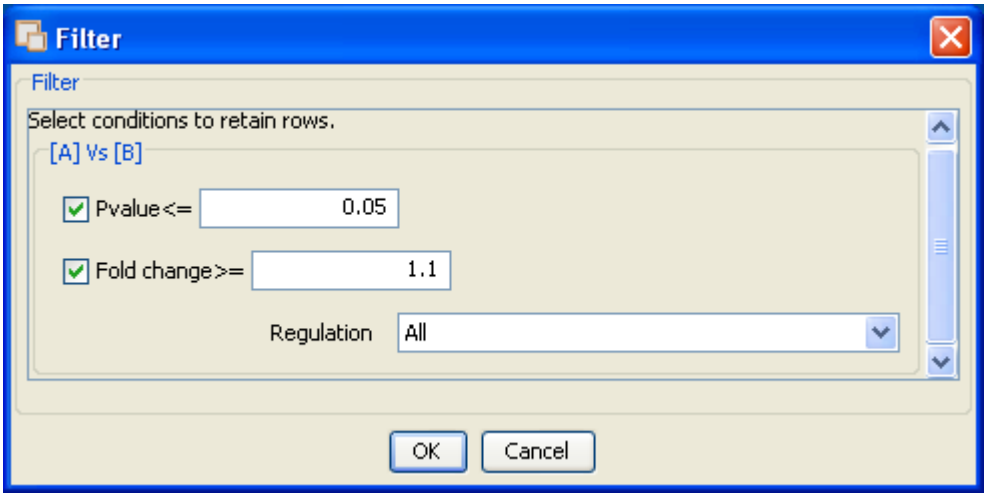


Figure 8.22: Filter on Significance Dialog

8.2.6 Clustering

The only clustering link available from the workflow browser is the K-Means which clusters the signal columns into 10 clusters. To run another algorithm or to change parameters, use the *Cluster* menu. See Section Clustering for more information.

8.2.7 Save Probeset List

Create Probeset List from Selection This link will create a probeset or Gene List from the selected genes. Normally, after identifying significantly expressed, you would like to save these genes or probesets of interest in the **ArrayAssist**. This will will save the selected probesets of genes as a gene list that will be available in any place in the tool. You will have to provide a name for the probeset or gene list and the mark to be used to associate with the list.

8.2.8 Import Gene Annotations

Once significant genes have been identified, you may want to explore the biology of the genes by bringing in annotations of the genes from a file, or annotating genes from various web sources via the annotation engine in

ArrayAssist. The following links allow you to import and fetch annotations into the dataset.

Importing Gene Annotations from Files. If you have your own set of gene annotations which you wish to import, prepare these annotations as a tab or comma separated file with genes as rows and annotation fields (name, symbol, locuslink etc.) as columns. Then import this file by going to the gene annotations dataset and using *Data* $\longrightarrow$*Columns*$\longrightarrow$*Import Columns*. Provide the file name and the gene identifier to be used for synchronizing columns in the file imported with columns in the gene annotations dataset. Next, mark each of the imported columns by setting the appropriate column mark in the *Data Properties* (appropriate marks include Unigene Id, Gene Name etc.). This will ensure two things: first, that these new columns are available from all child datasets, and second, that these columns are interpreted correctly by the annotation modules (web spidering, GO Browsing etc).

Marking Gene Annotations. Newly imported columns need to be marked by the type of annotation they carry (e.g., Genbank Accession etc). This can be done via *Data* $\longrightarrow$*Data Properties*. Marking the Gene Ontology Accession column is a prerequisite for GO Browsing as described below.

Fetching Gene Annotations from Web Sources. You can fetch annotations for selected genes from various public web sources. Select the genes of interest from any dataset or view, then choose the gene annotations dataset on the Navigator and click on this link. Select the public source of your interest, and indicate the input gene identifier you wish to start with (Unigene, Genbank Accession etc) and the information you need to fetch (gene name, alias etc). The information fetched will be updated in the gene annotations dataset or appended in some cases when the column fetched is not already there in the dataset. Note that the input identifiers used need to be marked (see Section Marking Annotation Columns), i.e., identified as Unigene, Genbank Accession etc. To mark a column, use *Data* $\longrightarrow$*Data Properties* and set the appropriate marks using the dropdown list provided for each column. Alternatively, the Annotation wizard has an option to mark columns. For more details on the public sites accessible and of the input and output identifiers, see Section Annotating Genes.

• Note that several marked gene annotation columns are hyperlinked, for instance the Probeset Id is linked to the Affymetrix NetAffx page, Gene Ontology accession is linked to the AMIGO page etc. For a list of these hyperlinks, see *File*$\longrightarrow$*Configuration*$\longrightarrow$*AffyURL*. These hyperlinks can be edited here.

8.2.9 Discovery Steps

This section contains links to dicover the biology of the selected genes by examining the GO terms associated with the selected genes or to visualize the location of the selected genes on the Chromosome viewer, if the gene location information is available in the dataset.

Gene Ontology Browsing. You can view *Gene Ontology* terms for the genes of interest in the Gene Ontology Browser invokable from this link. This browser offers several queries, a few of which are detailed below. See Section on GO Browser for a more complete description.

NOTE: To launch the GO browser, your currently active dataset needs to contain a Gene Ontology Accession column and this must be marked as such a column via *Data* $\longrightarrow$*Properties*. Each cell in this column should be a pipe separated list of GO terms, e.g., GO:0006118|GO:0005783|GO:0005792|GO:0016020.

- To view GO Terms for genes of interest and to identify enriched GO Terms, select genes of interest from any view and then click on the Find Go Terms with Significance  icon.

  Next move to the *Matched Tree view*. Here you will see all Gene Ontology terms associated with at least one of the genes along with their associated enrichment p-value (see Section GO Computation for details on how this is computed). You can navigate through this tree to identify GO Terms of interest.
- A tabular view of the p-values can also be obtained by clicking on the p-value Dataset  icon. This will produce a table in which rows are the above visible GO terms, and the columns contain various statistics (i.e., enrichment p-value, the number of genes having a particular GO term in the entire array, the number of genes amongst those selected having a particular GO term etc.).

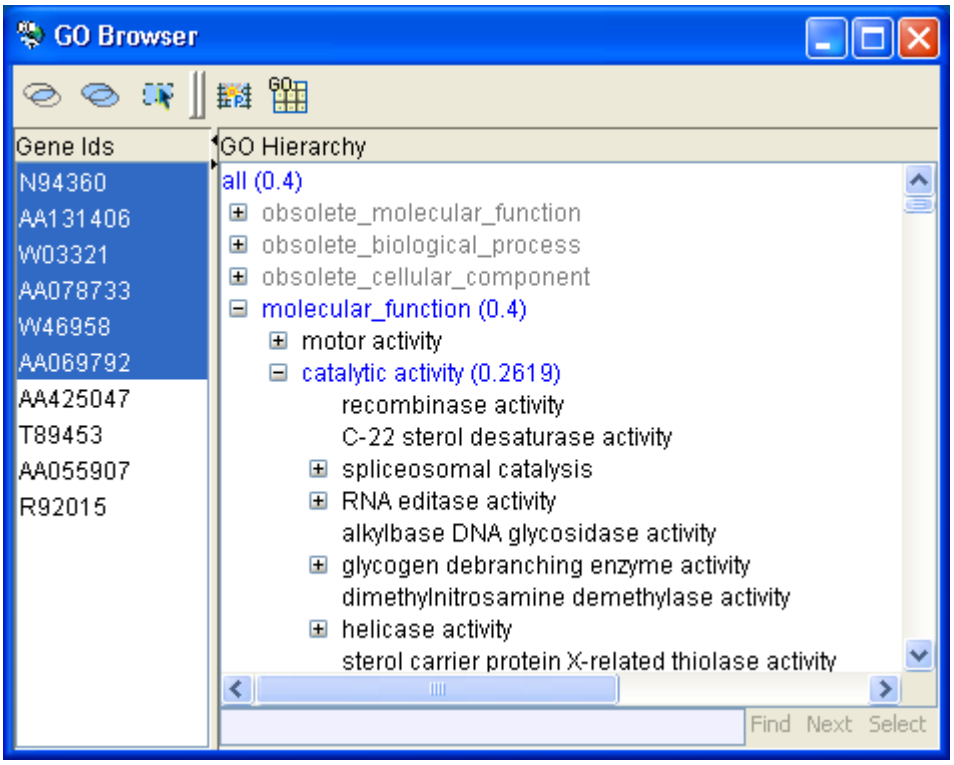


Figure 8.23: GO Browser

- Another tabular dataset can be obtained by clicking on the Gene Vs GO Dataset  icon and providing a cut-off p-value. This dataset shows probesets along the rows and GO Terms which occur in at least one of these probesets along the columns, with each cell being 0 or 1 indicating the presence or absence of that GO term for that probeset. This view is best viewed as a HeatMap by selecting the relevant columns and launching the HeatMap view from the *View* menu.
- You can also begin with a GO term (select it in the Full Hierarchy tab, if necessary you can use the search function to locate the term), and then click on Find All Genes with this Term  icon. This will select all probesets having this particular GO term in all the views and datasets.

Viewing Chromosomal Locations. Click on this link to view a scatter plot between Chromosome Number and Chromosome Start Location. Each probeset is depicted by a thin vertical line. Each chromosome is represented by a horizontal bar. Each probeset can be given a color as well. For instance, to color probesets by their fold changes or p-values, go to the Statistics output dataset in the Navigator and then launch the Chromosome Viewer. Use Right-Click *Properties* to color by the p-value or fold change columns.

NOTE: To launch the chromosome viewer, your currently active dataset needs to contain a Chromosome start location column and a Chromosome number column and this must be marked as such via *Data* $\longrightarrow$*Properties*.

Creating Custom Links. You can cause entries in a particular column to be treated as hyperlinks by changing the column mark to URL in *Data* $\longrightarrow$*Data Properties*. Subsequently, clicking on an entry in this column (either in the spreadsheet or in the lasso) will open the corresponding link in an external browser. Note that the entries in this column must be hyperlinks (i.e., of the form http:// etc.).

In case you wish to create a new hyperlink column, use the *Data*$\longrightarrow$*Column* $\longrightarrow$*Append Columns By Formula* command to create an appropriate string column and then use *Data* $\longrightarrow$*Data Properties* to mark this column as a URL column. For more details on creating new columns with formulae, see Section GO Computation.

8.2.10 Genome Browser

Genome Browser The Genome Browser can be invoked using this link. This browser allows viewing of several static prepackaged tracks. In addition, new tracks can be created based on currently open datasets. For more details on usage, see Section The Genome Browser.

Chapter 9

Analyzing Two-Dye Data

ArrayAssist can access and analyze files obtained by image analysis of most Two-Dye array formats with the following properties.

- There is usually one data file per experiment containing all spot quantified data for that experiment. Both Cy3 and Cy5 channel data are present in one file.
- The actual spot data in the data file is in tabular form, i.e., it is laid out as rows and columns, typically one row per spot with columns corresponding to various spot properties like gene name, block location, subblock location, foreground mean/median intensity, background mean/median intensity, etc.
- The tabular portion of the file could be only a part of the file and could be preceded by several lines containing additional experiment annotation details and possibly followed by several such lines as well.

Import of two-dye array formats happens via the two step process below.

Create Import Template. First, you need an *Import Template* for the specific files of your interest. **ArrayAssist** comes prepackaged with templates for the following file formats:

- **GenePix30**
- **Genepix40**
- **Genepix41**and
- **Imagene**

If you are working with one of these formats, try the appropriate template first by going through the *File* $\longrightarrow$*New Two-Dye Project* wizard. If it does not work (which might happen because of version differences) or if you are working with some other format, then you have two choices.

- Build your own template. This can be done for most formats which have data corresponding to one experiment in each file. See the description in Section The Two Dye Import Wizard for details.
- Seek **ArrayAssist** support for building the template. Send mail to `techservices@stratagene.com`and provide two sample files which you wish to import. We will send you a new template which will enable you to import your files into **ArrayAssist**.

Note that you *cannot* build your own templates for **Imagene** formats which have two separate files for Cy3 and Cy5. In addition, usage of the prepackaged Imagene formats currently have the following constraint: pairs of input files for each two-color array should have names _Cy3 and _Cy5 in there names with the portions before the underscore being identical.

Run Analysis. Second, import the files using this template and use the menu and workflow browser operations to proceed with the analysis. To perform the import, use the *File* $\longrightarrow$*New Two-Dye Project*. This will launch a wizard; choose the files of interest and provide the template name. See Section The Two Dye Workflow for details on further analysis.

9.1 The Two Dye Import Wizard

Step 1 - Select Files Use the *Choose File(s)* option on the wizard to locate the files of interest. Use this multiple times to locate files from different locations. *Remove file(s)* option can be used to remove selected files.

Step 2 - Select Template Use the *Select a template* drop down menu option to check if the format of interest is prepackaged. If not, use the *None* option and use the easy template building steps to create a template for the data. The template can be then saved. This template once created will become part of the drop down menu option and will be available from the next time.

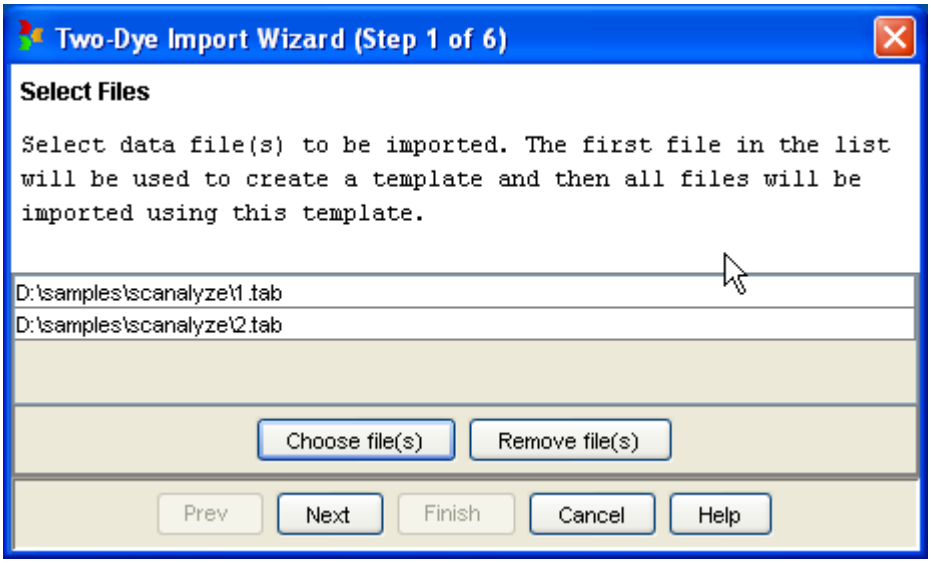

Two-Dye Import Wizard (Step 1 of 6)
Select Files
Select data file(s) to be imported. The first file in the list will be used to create a template and then all files will be imported using this template.
D:\samples\scanalyze\1.tab
D:\samples\scanalyze\2.tab
Choose file(s)
Remove file(s)
Prev
Next
Finish
Cancel
Help

Figure 9.1: Step 1 of Import Wizard

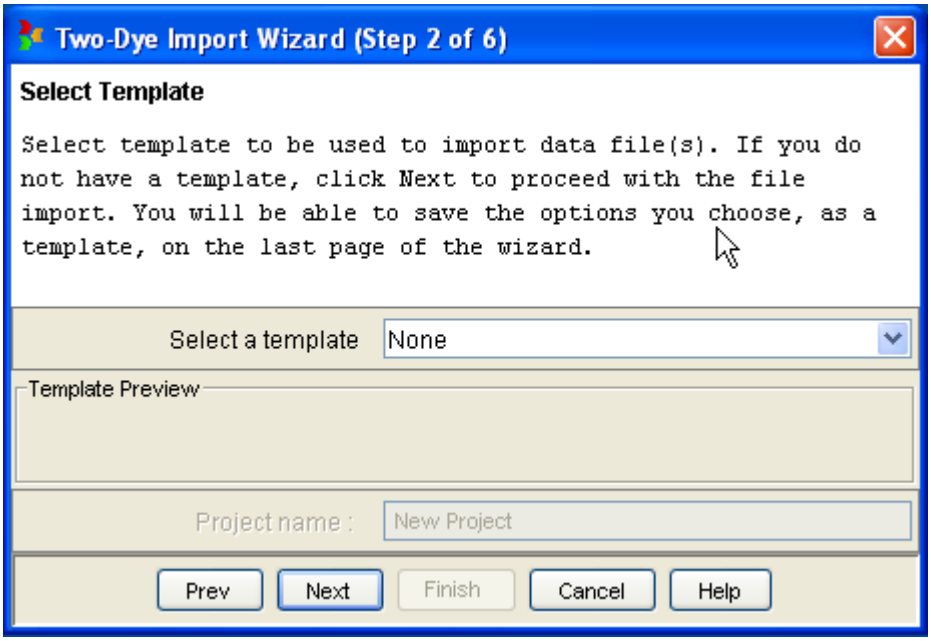

Two-Dye Import Wizard (Step 2 of 6)
Select Template
Select template to be used to import data file(s). If you do not have a template, click Next to proceed with the file import. You will be able to save the options you choose, as a template, on the last page of the wizard.
Select a template
None
Template Preview
Project name :
New Project
Prev
Next
Finish
Cancel
Help

Figure 9.2: Step 2 of Import Wizard

Step 3 - Format Options Use this step to specify the exact format of the data being brought in. Use the *Separator* option to specify the type of file. Use the *Text qualifier* to specify any special qualifiers used in the data file. Similarly use the *Missing value indicator* and *Comment indicator* to define the format of the text file.

The Separator separates fields in the file to be imported and is usually a tab, comma or space; new separators can be defined by scrolling down to EnterNew and providing the appropriate symbol in the textbox.

The Text Indicator is usually just inverted commas (") used to ignore separators which appear within text strings.

The Missing Value Indicator indicates the symbol(s), if any, used to represent a missing value in the file. This applies only to cases where the value is represented explicitly by a symbol such as N/A, NA or —.

Comment Indicators are markers at the beginning of the line which indicate that the line should be skipped (typical examples is the # symbol).

Step 4 - Select row scope for import The purpose of this step is to identify which rows need to be imported. The rows to be imported must be contiguous in the file. The rules defined for importing rows from this file will then apply to all other files to be imported. Choose one of three options below.

The default option is to select all rows in the file. Alternatively, you can choose to take rows from a specific row number to a specific row number (use the preview window to identify row numbers) by entering the row numbers in the appropriate textboxes. Remember to press the enter key before proceeding. In addition, for situations where the data of interest lies between specific text markers, e.g., Begin Data and End Data, use option 3 to specify these markers; these markers must appear at the very beginning of their respective lines and the actual data starts from the line after the first marker and ends on the line preceding the second marker. Note also that instead of choosing one of the options from the radio buttons, you can choose to select specific contiguous rows from the preview window itself by using Left-Click and Shift-Left-Click on the row header.

The panel at the bottom asks you to indicate whether or not there is a header row; in the latter case, dummy column names will be assigned.

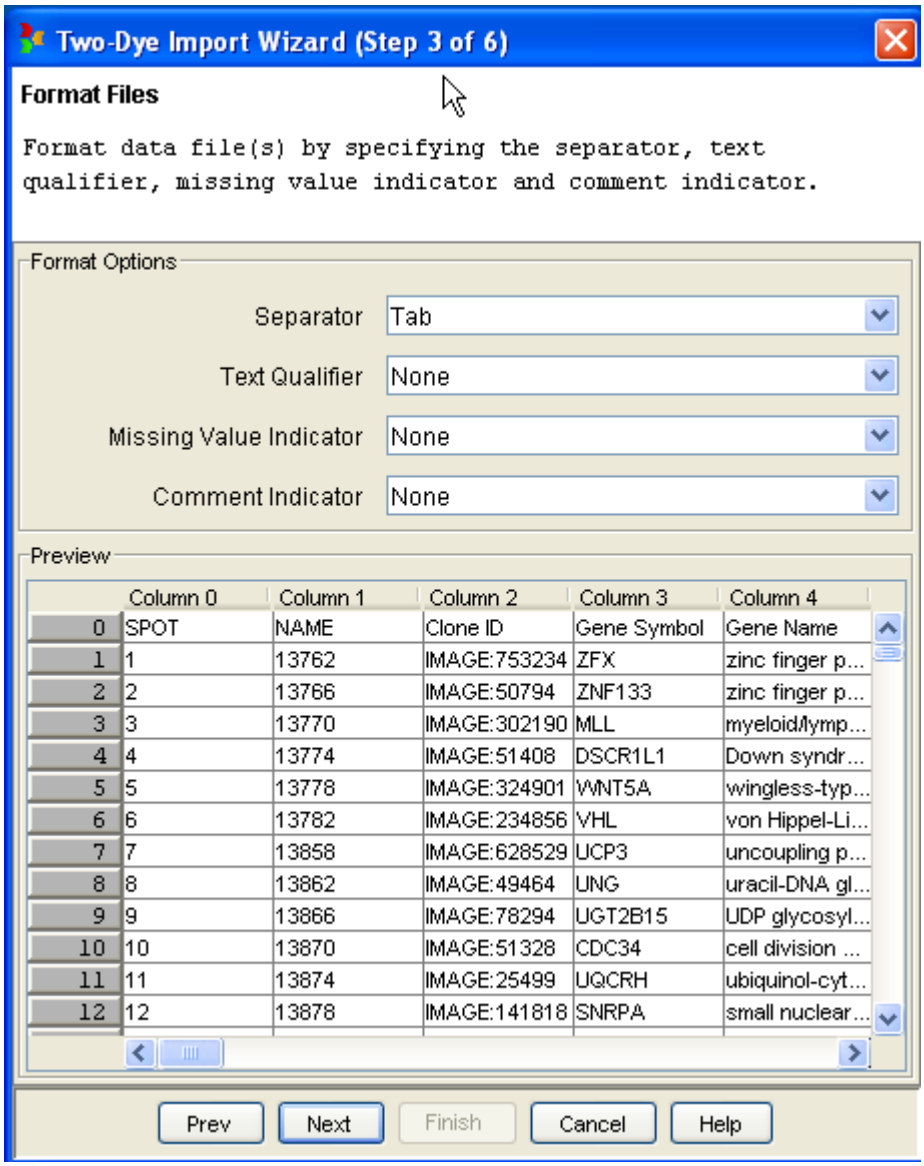

Two-Dye Import Wizard (Step 3 of 6)
Format Files
Format data file(s) by specifying the separator, text qualifier, missing value indicator and comment indicator.
Format Options
Separator
Tab
Text Qualifier
None
Missing Value Indicator
None
Comment Indicator
None
Preview
Column 0 Column 1 Column 2 Column 3 Column 4
0 SPOT NAME Clone ID Gene Symbol Gene Name
1 1 13762 IMAGE:753234 ZFX zinc finger p...
2 2 13766 IMAGE:50794 ZNF133 zinc finger p...
3 3 13770 IMAGE:302190 MLL myeloid/lymp...
4 4 13774 IMAGE:51408 DSCR1L1 Down syndr...
5 5 13778 IMAGE:324901 WNT5A wingless-typ...
6 6 13782 IMAGE:234856 VHL von Hippel-Li...
7 7 13858 IMAGE:628529 UCP3 uncoupling p...
8 8 13862 IMAGE:49464 UNG uracil-DNA gl...
9 9 13866 IMAGE:78294 UGT2B15 UDP glycosyl...
10 10 13870 IMAGE:51328 CDC34 cell division ...
11 11 13874 IMAGE:25499 UQCRH ubiquinol-cyt...
12 12 13878 IMAGE:141818 SNRPA small nuclear...
Prev
Next
Finish
Cancel
Help

Figure 9.3: Step 3 of Import Wizard

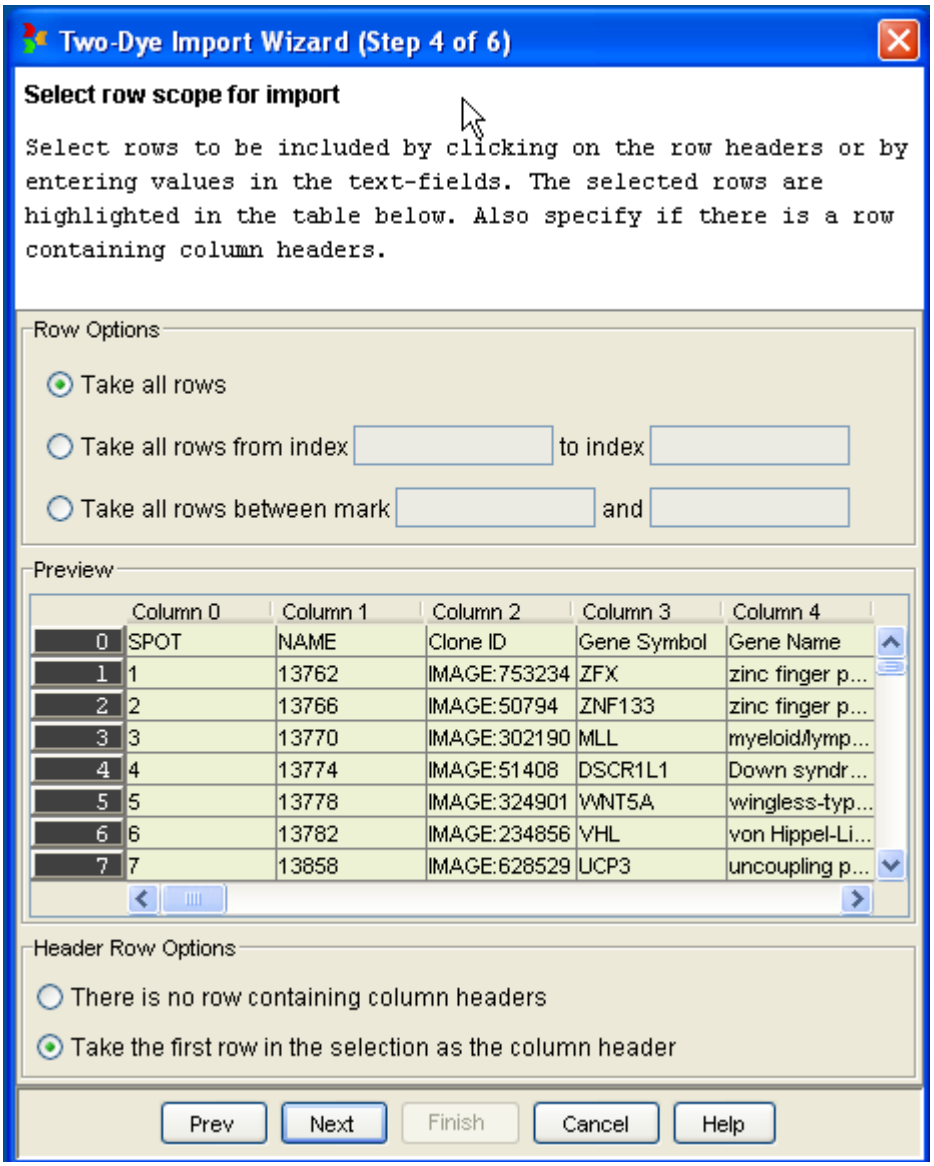

Two-Dye Import Wizard (Step 4 of 6)
Select row scope for import
Select rows to be included by clicking on the row headers or by entering values in the text-fields. The selected rows are highlighted in the table below. Also specify if there is a row containing column headers.
Row Options
Take all rows
Take all rows from index
to index
Take all rows between mark
and
Preview
Column 0 Column 1 Column 2 Column 3 Column 4
0 SPOT NAME Clone ID Gene Symbol Gene Name
1 1 13762 IMAGE:753234 ZFX zinc finger p...
2 2 13766 IMAGE:50794 ZNF133 zinc finger p...
3 3 13770 IMAGE:302190 MLL myeloid/lymp...
4 4 13774 IMAGE:51408 DSCR1L1 Down syndr...
5 5 13778 IMAGE:324901 WNT5A wingless-typ...
6 6 13782 IMAGE:234856 VHL von Hippel-Li...
7 7 13858 IMAGE:628529 UCP3 uncoupling p...
Header Row Options
There is no row containing column headers
Take the first row in the selection as the column header
Prev
Next
Finish
Cancel
Help

Figure 9.4: Step 4 of Import Wizard

Step 5 - Column Options and Column Marks The purpose of this step is to identify which columns are to be imported and what the type of each column is. The rules defined for importing rows from this file will then apply to all other files to be imported.

Select which columns need to be imported by checking/unchecking the textboxes on the left which appear against each column. In Column Options, specify how the columns selected by this procedure will be identified in other files to be imported; this identification can be done either by using the same *column names* or by using the same *column numbers*. The "column number" option is safer in instances where the actual column name could change from file to file, maybe due to addition of a date or the filename to the column name.

The Merge Options at the bottom specify how multiple files imported should be merged. Use the *alignment by row identifiers* option if the order of appearance of rows is not identical in all the files, and choose the *alignment by order of occurrence* otherwise. In the former case, you will need to mark one of the columns as an Identifier Column, as described below.

The most detailed task on this page is to provide a *Mark* for each column. The marks appear in the dropdown obtained by clicking on the None in the Column Mark panel against the relevant column. The set of available marks is listed below, with a brief explanation on what each mark means. Of these, only the Signals marks are compulsory.

Step 5 of the wizard requires identification of Column Marks. Marks along with Tags that are generated by **ArrayAssist** are used intelligently by the workflow browser to carry out the analysis. Tags and Marks are explained in detail below. The Column Mark column gives a drop down menu option to choose and match the data with the appropriate mark.

A Mark is associated with each spot property/data point being imported into the **ArrayAssist** spreadsheet. The broad categories of Marks are as follows:

- Signal Values
- The Spot Identifier and Coordinates Marks
- The Spot Type and Quality Marks
- Gene Annotation information

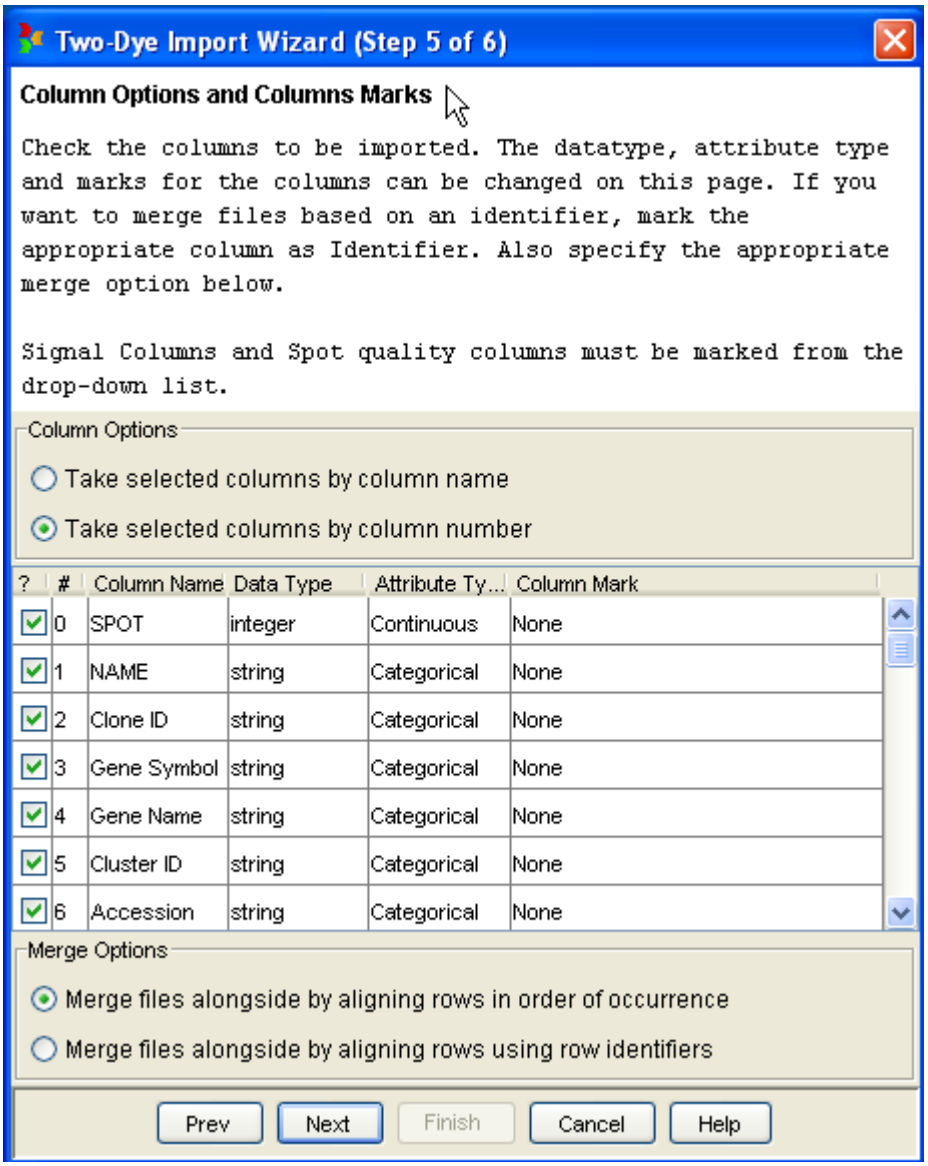

Two-Dye Import Wizard (Step 5 of 6)
Column Options and Columns Marks
Check the columns to be imported. The datatype, attribute type and marks for the columns can be changed on this page. If you want to merge files based on an identifier, mark the appropriate column as Identifier. Also specify the appropriate merge option below.
Signal Columns and Spot quality columns must be marked from the drop-down list.
Column Options
Take selected columns by column name
Take selected columns by column number
? # Column Name Data Type Attribute Ty... Column Mark
0 SPOT integer Continuous None
1 NAME string Categorical None
2 Clone ID string Categorical None
3 Gene Symbol string Categorical None
4 Gene Name string Categorical None
5 Cluster ID string Categorical None
6 Accession string Categorical None
Merge Options
Merge files alongside by aligning rows in order of occurrence
Merge files alongside by aligning rows using row identifiers
Prev
Next
Finish
Cancel
Help

Figure 9.5: Step 5 of Import Wizard

Associating data columns with Column Marks. This step asks for associating column names in the files with standard quantities associated with two-dye analysis. A list and explanation of these quantities appears below. Cretain columns are mandatory for a two-dye project, like the signal columns. For the remaining quantities, associating column marks is optional but may be useful for later steps, e.g., filtering, normalization etc. To associate a column with a quantity use the drop down menu.

Two warning notes are shown by **ArrayAssist** if there is no data associated with either Spot type or Flags. These messages are just for information. Flag is a quality parameter generated by the image analysis software. Spot type refers to specific controls like housekeeping genes, spike in genes, negative control genes etc.

- **Foreground intensities of Cy3/Channel 1 and Cy5/Channel 2:** There could be multiple columns corresponding to the foreground intensity in the input files, e.g., mean foreground intensity or median foreground intensity; in such cases the median intensity is recommended over the mean intensity.
- **Background intensities of Cy3/Channel 1 and Cy5/Channel 2:** There could be multiple columns, corresponding to the background intensity in the input files, e.g., mean background intensity or median background intensity; in such cases, the median intensity is recommended over the mean intensity. Typically, the same type of signal should be used for both background and foreground intensities. If foreground intensity is specified, the it is mandatory to mark the background intensity columns.
- **Background Corrected Intensities for Cy3/Channel 1 and Cy5/Channel 2:** Some scanners will directly output background corrected intensities and call then the signal column. Normally, the file header my specify the background correction used. If these columns are available they should be markes as background corrected signal columns.
- **Normalized Background Corrected intensities of Cy3/Channel 1 and Cy5/Channel 2:** Some scanners and output formats would output a normalized background corrected signal values. If these are present, such a column can be marked and will be brought into the dataset.

- **Normalized Background Corrected ratios:** Certain scanners and output formats will directly output normalized backgroud corrected ratio signals. If these are present, such a column can be marked and will be brought into the dataset.
- **Normalized Background Corrected Cy5/Cy3 log ratios:** Certain scanners and output formats will directly output normalized backgroud corrected log ratio signals. If these are present, such a column can be marked and will be brought into the dataset.
- **Identifier:** This is the row identifier in the dataset. If this is a unique column in the file, and identifies the gene or spot on the array, then the Identifier columns can be used to merge multiple files together. Certain scanner output formats or arrays may not output all the spots in the same order. Then the Identifier column must be used to merge multiple files or arrays and brought into **ArrayAssist** by explicitly chosing the option to merge files alongdside by aligning rows using the row Identifiers in the merge option at the bottom of the page.
- **Spot Identifier:** This is an optional field. Each spot typically has a spot number on the chip. If the spot identifier is used to merge rows, then this column must be marked as an Identifier column.
- **Physical X and Y Spot Coordinates:** These are optional and are required to view a physical image of the chip via scatter plots in **ArrayAssist**.
- **Block Number(s):** Typically, spotted arrays are spotted in blocks. These blocks are numbered either with block-row and block-column numbers or with single numbers from 1 to the number of blocks; select one of these two options. This field is optional but useful if you want to normalize data in each block separately.
- **Flags:** Each spot has an associated flag which can be turned on in the image analysis step to indicate that the spot is bad. These flags will be useful for filtering spots.
- **Spot p-value:** Some Image analysis software output a p-value based on the error model used in the computation of each log ratio.
- **Gene Description:** The purpose of this is purely to carry over gene description information to the output dataset.

- **Other Annotation Marks:** If the dataset contains other annotation columns like the GeneBank Accession Numner, the Gene Name, etc, these columns can be marked on the dataset while importing data into **ArrayAssist**. If the dataset contains such annotation columns, they can used for running the annotation workflow or launching the genome browser.
- **Duplicate and New Marks:** Other than signals, **ArrayAssist** will not allow the same mark to be used for multiple columns. New marks can be defined by choosing the EnterNew towards the bottom of the marks dropdown list; however, filtering based on newly defined marks will not be possible via the current workflow steps and will need to be performed manually, i.e., using the filter utility or by writing a script etc.

Tags are associated with various forms of raw data and comprise of the following. Depending upon the the columns that are marked in the input files, datasets corresponding to the vaious tages will be automatically created in the project.

- Raw Signals of Cy3 and Cy5 - Foreground and Background
- Background corrected signal of Cy3 and Cy5
- Normalized signal values of Cy3 and Cy5
- Signal ratio of Cy3 and Cy5
- Log Signal ratio of Cy3 and Cy5
- Dye swapped data, if relevant

NOTE: All panels and the whole window is resizable by dragging if needed. Also if Spot Type or Flag is not marked then a warning is issued before proceeding.

Step 6 - Summary This step shows a summary of all the options chosen for building the template. Use the *Template name* to provide a name for this template. The template will be saved and can be subsequently used to import other files that have the same format. Use the *Project name* option to provide a name for the project being created.

This is the last step in the wizard, choose *Finish* to bring the data into **ArrayAssist** for further analysis using the Workflow Browser.

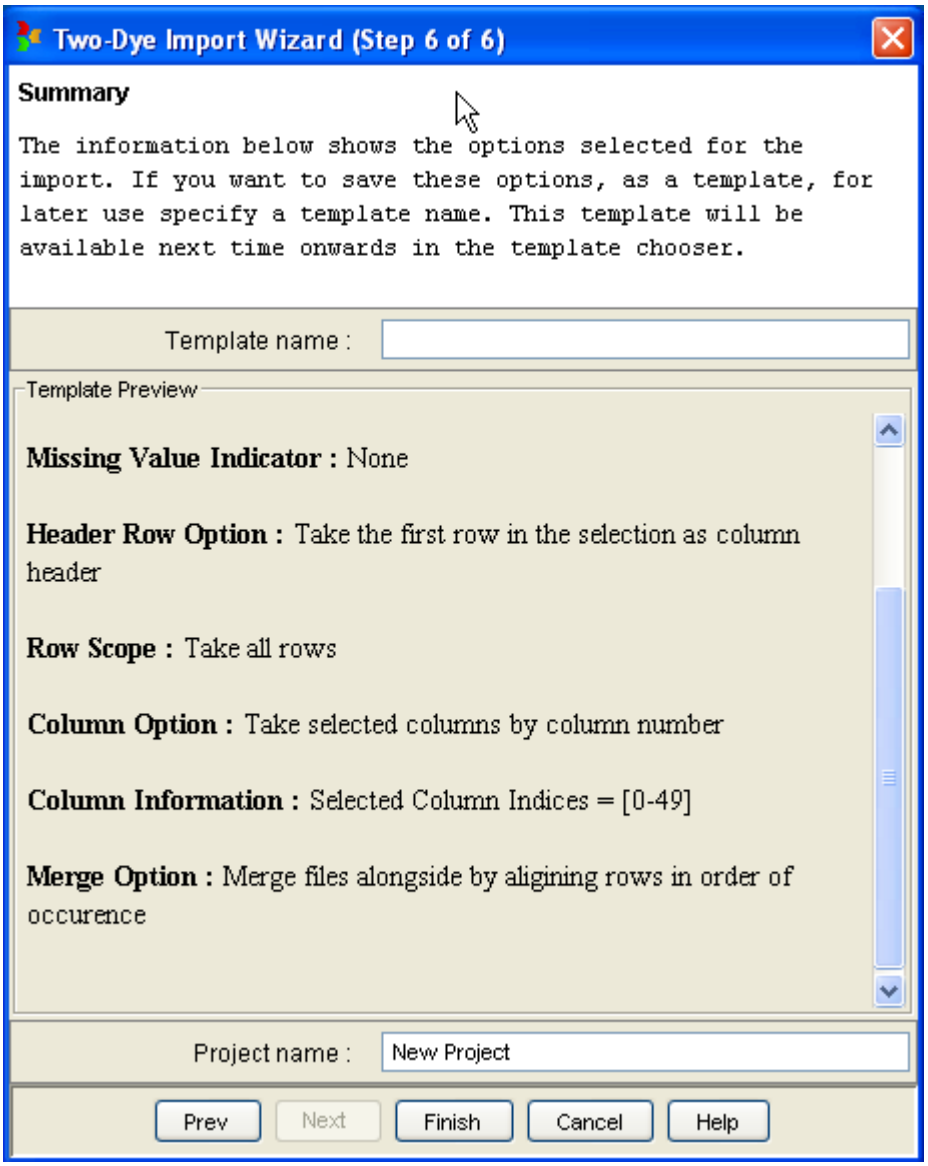

Two-Dye Import Wizard (Step 6 of 6)
Summary
The information below shows the options selected for the import. If you want to save these options, as a template, for later use specify a template name. This template will be available next time onwards in the template chooser.
Template name :
Template Preview
Missing Value Indicator : None
Header Row Option : Take the first row in the selection as column header
Row Scope : Take all rows
Column Option : Take selected columns by column number
Column Information : Selected Column Indices = [0-49]
Merge Option : Merge files alongside by aligining rows in order of occurence
Project name :
New Project
Prev
Next
Finish
Cancel
Help

Figure 9.6: Step 6 of Import Wizard

Once the two-dye data is loaded into **ArrayAssist**, a normal analysis flow can be performed by the use of the workflow browser. The steps in the workflow browser captures the most common two-dye analysis workflow.

NOTE: If the import wizard returns with an error, then there is a mismatch between the template used and the files input. Please send mail to techservices@stratagene.comwith a description of the error message along with one or two sample files.

9.2 The Two Dye Workflow

After creating the appropriate template, use *File* $\longrightarrow$*Import SingleDye* wizard to import files using this template. Select the files of interest and select the template from the drop-down list of all templates. Successful import now will result in the creation of a new single-dye project. The navigator on the left should show the number of rows in the project (which corresponds to the number of probes on one array) and the number of columns (which includes all type of signals, flags and ids).

The Initial Datasets. In addition, the navigator should show either a Raw dataset, a BG (background) Corrected dataset, or a Normalized BG Corrected dataset. More than one of these datasets could also be shown depending upon which type of signals were marked in the template creation process. If Foreground and Background Signals were marked then a raw dataset containing foreground and background values for each array imported will be shown, and likewise, for Background Corrected and Normalized signal values. In addition to the signal columns, all these datasets will contain all other columns marked in the template creation process. The list of columns and their types and marks can be seen using Data Properties  icon. If you used a template that came prepackaged with **ArrayAssist**, then you may not be familiar with the notion of *column marks*; refer to Section Column Options and Marks for details.

NOTE: If the navigator does not show any of Raw, BG Corrected or Normalized, then the template used for import did not have signals marked correctly. Go back and create a new template making sure that signal columns are marked appropriately this time or send emailx to techservices@stratagene.comto request support.

NOTE: Most datasets and views in **ArrayAssist** are *lassoed*, i.e., selecting one or more rows/columns/points will highlight the corresponding rows/columns/points in all other datasets and views. In addition, if you select probes from any dataset or view, signal values and gene annotations for the selected probes can be viewed using *View* $\longrightarrow$*Lasso* (you may need to customize the columns visible on the Lasso view using Right-Click *Properties*).

The Workflow. Once the project opens up with the appropriate datasets in the navigator then the primary analysis steps are enumerated in the workflow browser panel on the right. These steps can be run by clicking upon the corresponding links. A listing and explanation of these steps appears in the sections below.

NOTE: Steps in the workflow browser are related to the dataset that is in focus in the navigator. Each step operates on the dataset in focus. Further, it may or may not be applicable to this dataset. Before running a specific step, you may need to move focus to the relevant dataset in the navigator.

9.2.1 Getting Started

Click on this link to take you to the chapter on Analyzing Two-Dye Data.

9.2.2 The Experiment Grouping

The very first step is providing Experiment Grouping. The *Experiment Grouping* view which comes up will initially just have the imported file names. The task of grouping will involve providing more columns to this view containing *Experiment Factor* and *Experiment Grouping* information. A Control vs. Treatment type experiment will have a single factor comprising 2 groups, Control and Treatment. A more complicated Two-Way experiment could feature two experiment factors, genotype and dosage, with genotype having transgenic and non-transgenic groups, and dosage having 5, 10, and 50mg groups. Adding, removing and editing Experiment Factors and associated groups can be performed using the icons described below.

Reading Factor and Grouping Information from Files. Click on the Read Factors,Groups from File  icon to read in all the Experiment Factor and Grouping information from a tab or comma separated text file. The file

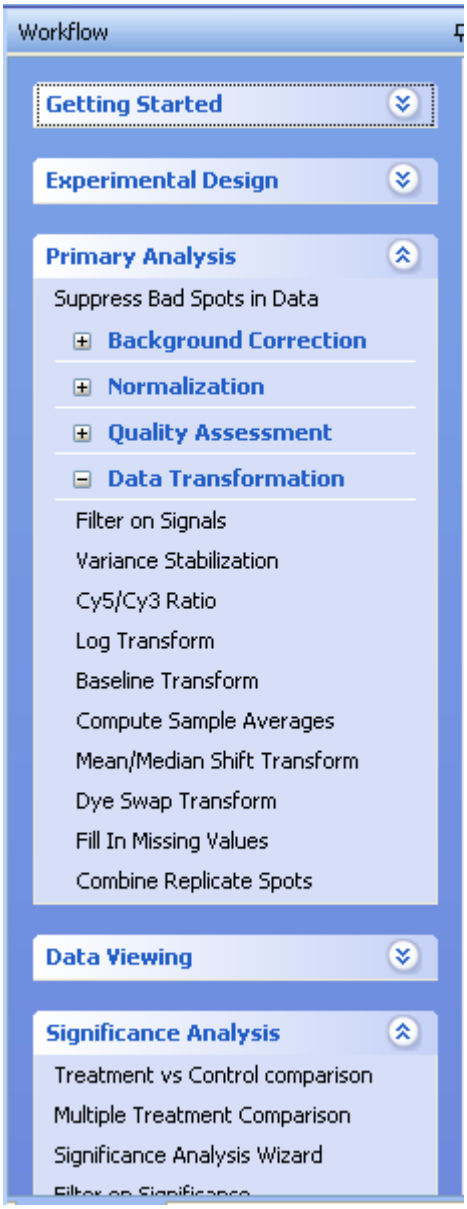


Figure 9.7: The Two-Dye Workflow Browser

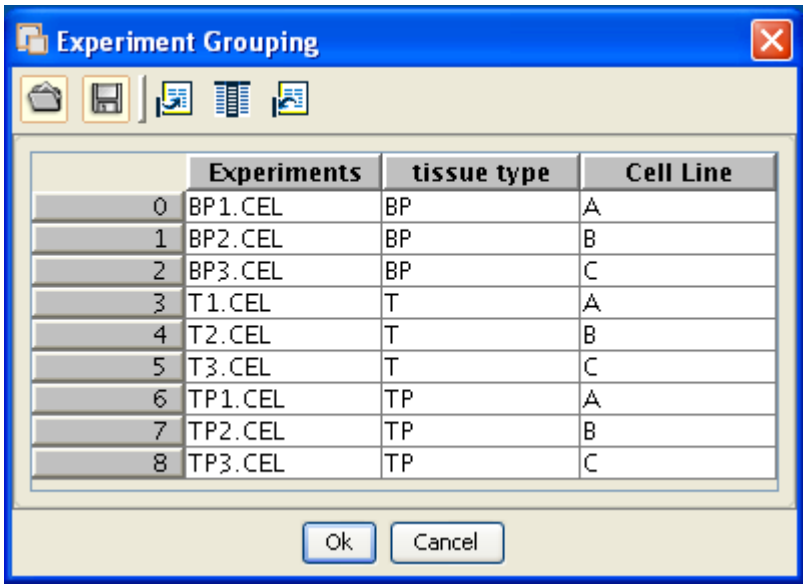


Figure 9.8: The Experiment Grouping View With Two Factors

should contain a column containing imported file names; in addition, it should have one column per factor containing the grouping information for that factor. Here is an example tab separated file. The result of reading this tab file in is the new columns corresponding to each factor in the Experiment Grouping view.

```
#comments
#comments
filename genotype dosage
A1.GPR    NT       0
A2.GPR     T       0
A3.GPR    NT      20
A4.GPR     T      20
A5.GPR    NT      50
A6.GPR     T      50
```

Adding a New Experiment Factor. Click on the Add Experiment Factor icon to create a new Experiment Factor and give it a name when prompted. This will show the following view asking for grouping information corresponding to the experiment factor at hand. The files shown in this

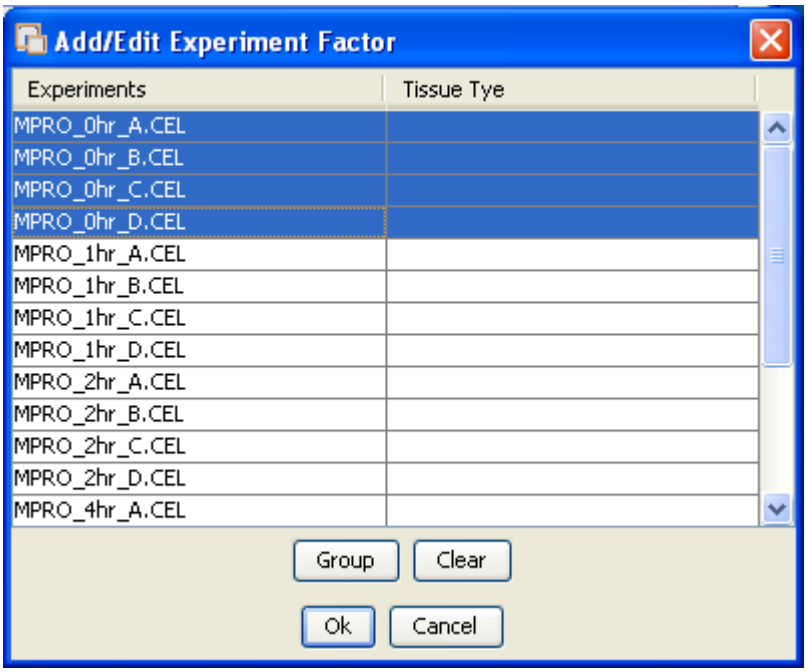


Figure 9.9: Specify Groups within an Experiment Factor

view need to be grouped, with each group comprising biological replicate arrays. To do this grouping, select a set of imported files, then click on the *Group* button, and provide a name for the group. Selecting files uses Left-Click , Ctrl-Left-Click , and Shift-Left-Click , as before.

Editing an Experiment Factor. Click on the Edit Experiment Factor icon to edit an Experiment Factor. This will pull up the same grouping interface described in the previous paragraph. The groups already set here can be changed on this page.

Remove an Experiment Factor. Click on the Remove Experiment Factor icon to remove an Experiment Factor.

9.2.3 Primary Analysis

This section includes links to do primary analysis of two-dye data. They include methods to supress bad spots in the data, vaious methods of background correction, normalization, quality assessment and data transformations. These are detailed below:

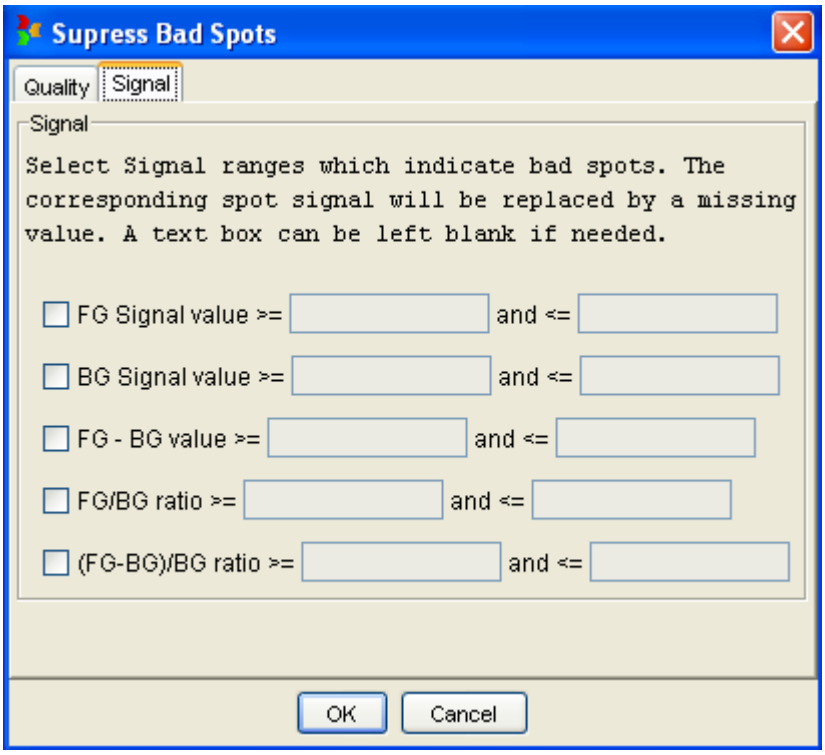


Figure 9.10: Suppress Bad Spots

Suppress Bad Spots in Data This is a quality control step and is optional. This link can be used to filter based on flags generated by the image analysis software or based on the signal values. Typically, low signal values are filtered to remove noise from the data. The pop up window has two tabs, one for filtering on flags and the other for filtering on signals.

This step will create a new dataset in which signal values corresponding to bad spots are replaced by missing values; all further operations can be performed on this dataset. Bad spots can be identified by quality marks The Spot Type and Quality Marks or by signal value ranges. The signal value used is the one present in the dataset that is in focus in the navigator.

Background Correction Once spots to be filtered have been identified, the next step is to perform background correction. Of course, this step

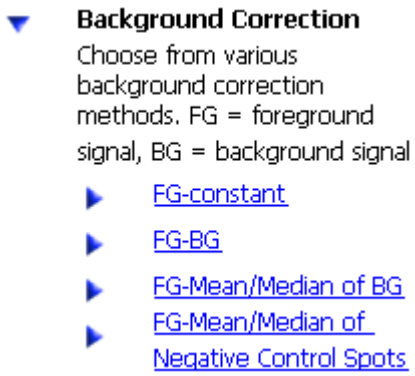

Background Correction
Choose from various background correction methods. FG = foreground signal, BG = background signal
FG-constant
FG-BG
FG-Mean/Median of BG
FG-Mean/Median of Negative Control Spots

Figure 9.11: Background Correction

is applicable only if start point was foreground and background intensities for each channel. If start point is data with already background corrected channel intensities or ratios or log-ratios, this option will not be applicable.

There are four choices for background correction

- Foreground - constant: This option can be used to subtract a constant value from all the foreground intensities. Select zero (0) if no correction needs to be done.
- FG-BG: This option is used to subtract background intensities from their respective foreground intensities.
- FG-Mean/Median of BG: This option is used to subtract either the mean or the median of the background from all foreground intensities for each channel on all arrays.
- FG-Mean/Median of Negative Control spots: This option is used to subtract either the mean or median of negative control spots from all foreground intensities for each channel on all arrays.

NOTE: If you did not mark any column as Spot Type while creating the template or if you wish to create and mark a new column containing negative control indicators as Spot Type, then select the probes of interest on the spreadsheet, use *Data* $\longrightarrow$*Row Operations* $\longrightarrow$*Label Rows* to label the negative control probes, then use **Data** $\longrightarrow$**Properties** to mark this newly added Label column as the Spot Type column.

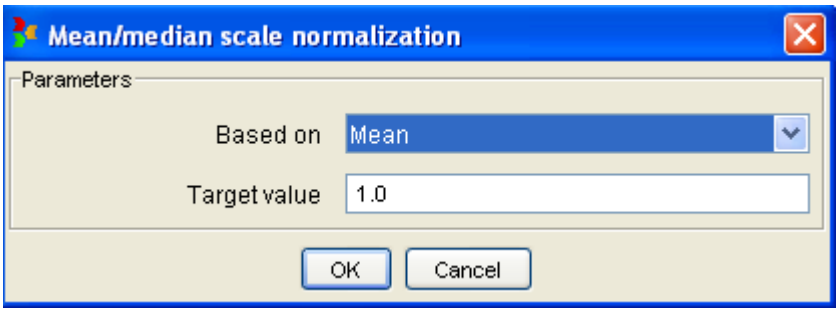


Figure 9.12: Normalization

NOTE: Background Correction could result in negative values, which could create problems later. You can suppress negative values using the Suppress Bad Spots link in the workflow browser; suppress spots where the background corrected signal is less than 0.

Normalization The next step in the analysis is normalization.

Normalization is admissible only on Background Corrected datasets. If for some reason you do not wish to perform background correction but wish to go on to normalization directly, then use the FG-constant background correction method with the constant set to 0 to derive a background corrected dataset.

- Mean/Median scale: The most common normalization method is to equalize the array means or medians by scaling (Mean/Median Scale Option); you will need to provide the target value which all medians/means attain after normalization.
- Mean/Median scale using Housekeeping genes: The Mean/Median scaling using Housekeeping genes option is useful in situations where most genes on the chip are changing is response to stimulus and therefore equalizing means/medians does not make sense. In this situation, the means/medians of housekeeping spots are equalized across chips by scaling. Housekeeping spots are identified using the Spot Type mark (as was the case for negative controls in background correction Background Correction).
- Lowess Cy5 against Cy3: This option asks for Lowess normalization for normalizing Cy5 against Cy3 on each array to remove

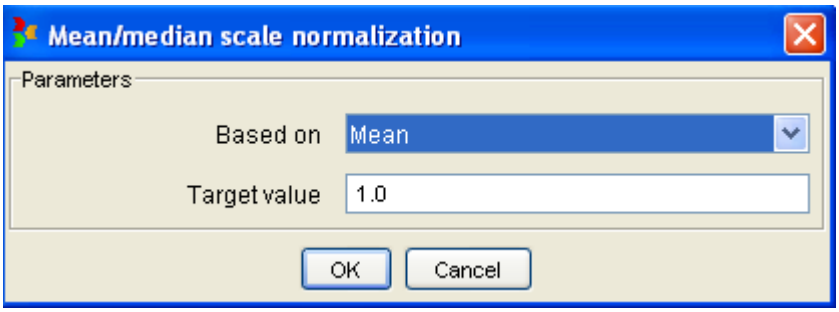


Figure 9.13: Normalization

differential dye effects. Lowess normalization is used if you believe that most genes are not differentially expressed between the two channels but differential dye effects can cause lot of genes to appear as differentially expressed. In this method, the MVA plot (mean versus difference plot) of the two channel values is plotted and a smooth curve is fit on this plot.

The advantage of Lowess over MeanShift is that Lowess is a more powerful method because of its ability to perform differential correction in different intensity ranges while MeanShift is much coarser; it uses the same correction everywhere.

Quality Assessment The quality assessment step has a few visualization options to check the quality of the data. This step can be used to decide the data points to carry forward for further analysis.

- **Cy5 Cy3 data quality plots:** This plot gives the MVA plot for the different arrays using the raw signal values for the two channels, Cy5 and Cy3.
- **Data quality matrix plots:** This is multi-scatter plot view of all the channels and all the arrays in one view. This uses the normalized data of Cy5 and Cy3 channels. This snapshot view gives a quick idea about the quality of the normalized data.
- **Principal Component Analysis on Arrays.** This link will perform principal component analysis on the arrays. It will show the standard PCA plots (see PCA for more details). The most relevant of these plots used to check data quality is the *PCA scores* plot, which shows one point per array and is colored by the

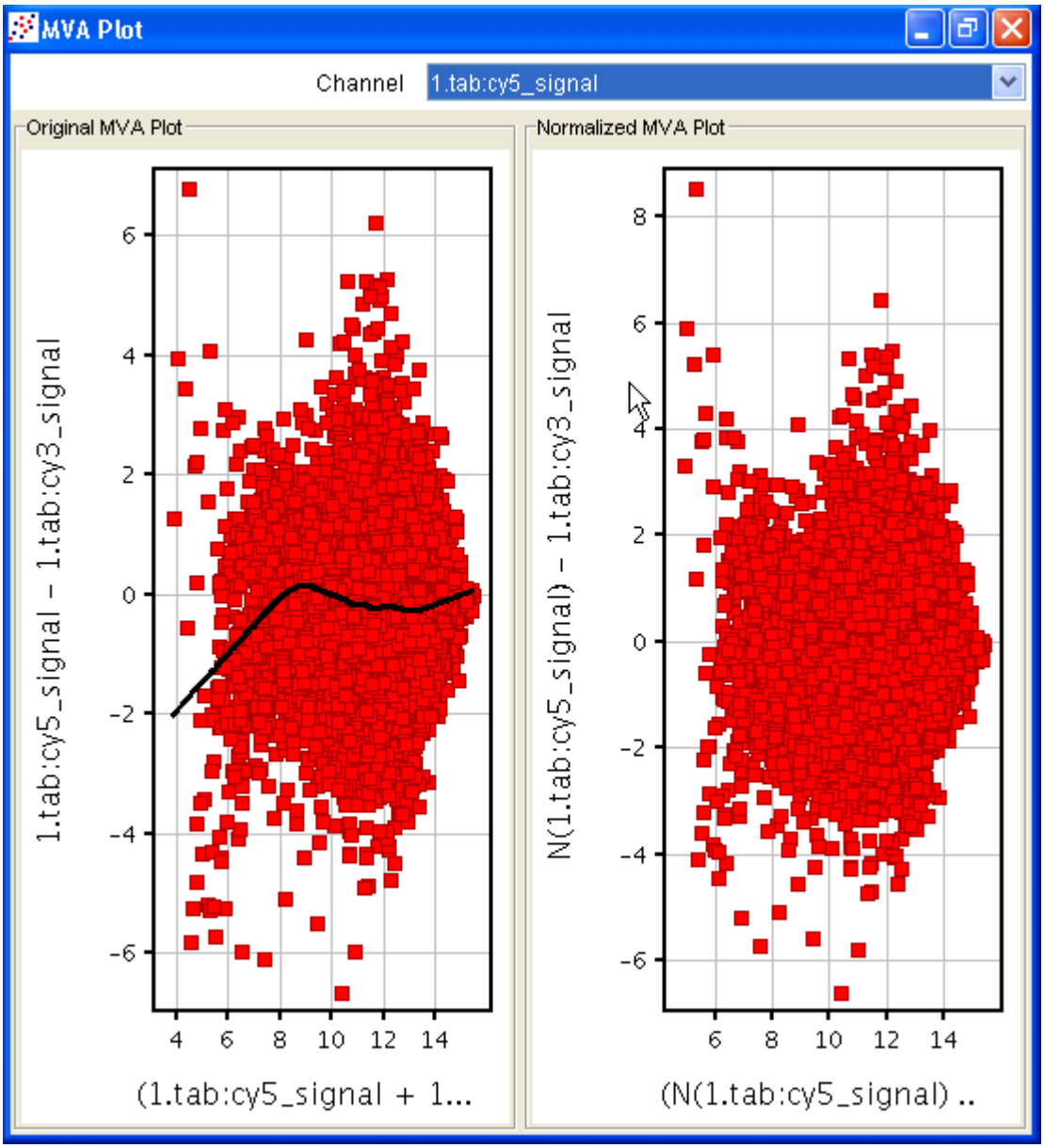


Figure 9.14: MVA Plot

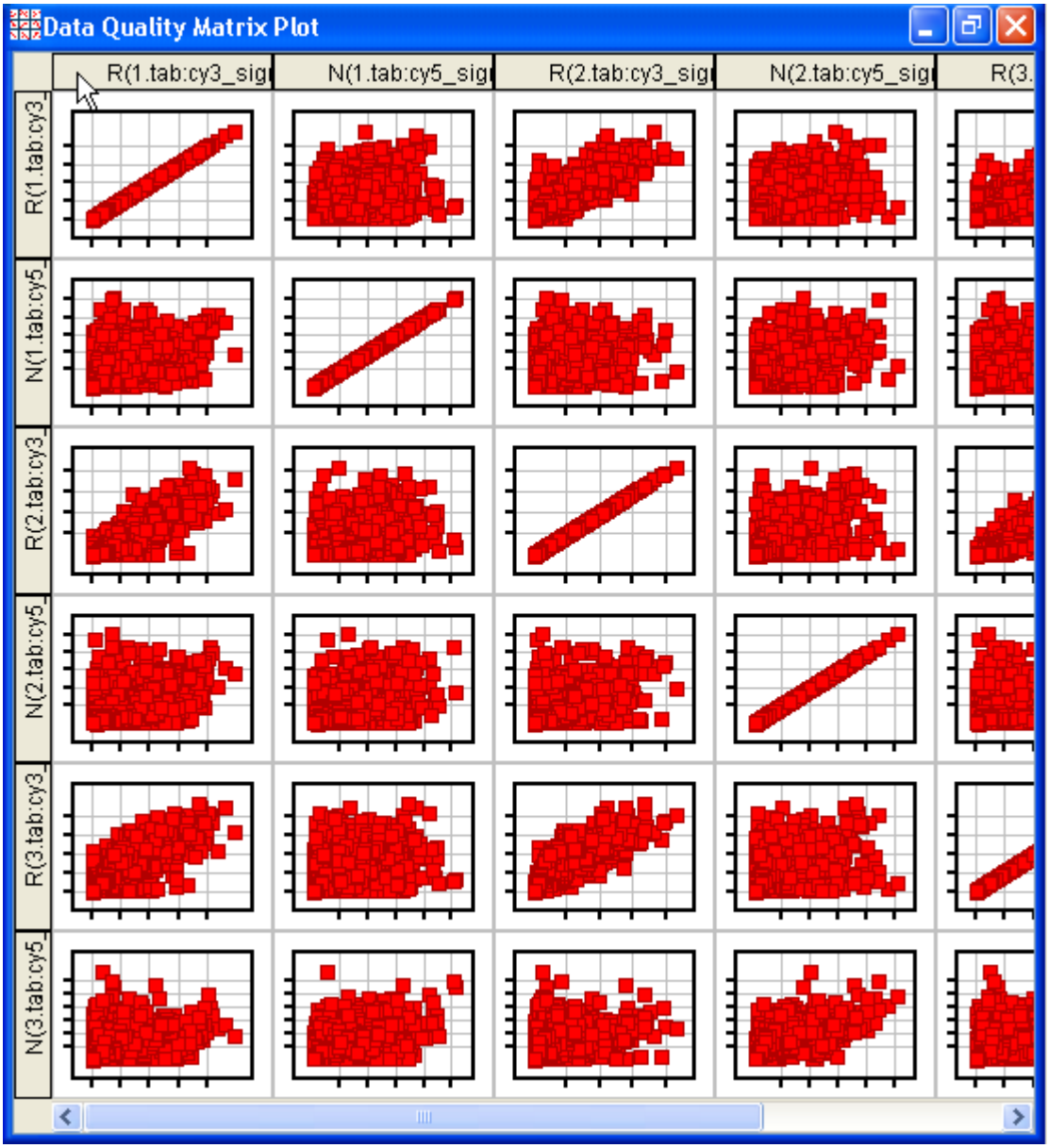


Figure 9.15: Matrix Plot

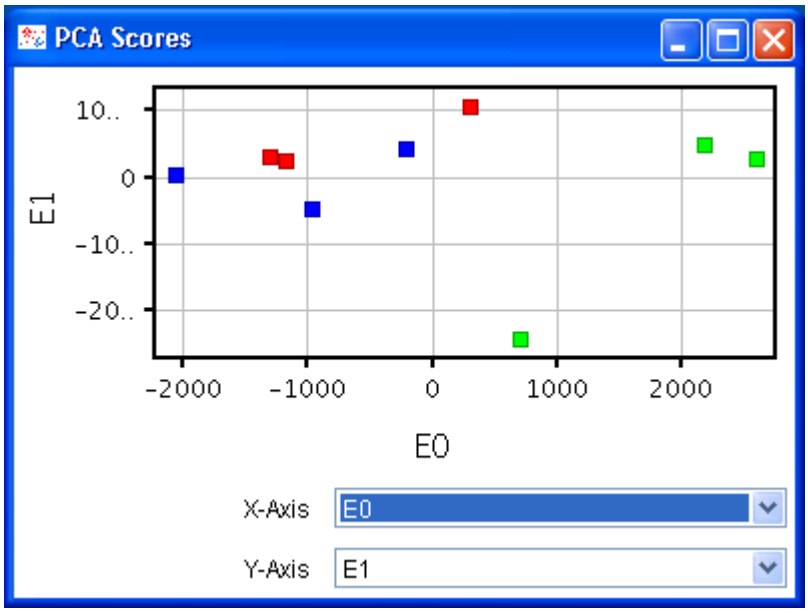


Figure 9.16: PCA Scores Showing Replicate Groups Separated

Experiment Factors provided earlier in the *Experiment Grouping* view. This allows viewing of separations between groups of replicates. Ideally, replicates within a group should cluster together and separately from arrays in other groups. The *PCA scores* plot can be color customized via Right-Click *Properties*. All the Experiment Factors should occur here, along with the Principal Components E0, E1 etc. The PCA Scores view is *lassoed*, i.e., selecting one or more points on this plot will highlight the corresponding columns (i.e., arrays) in all the datasets and views. Further details on running PCA appear in Section PCA.

Data Transformation Once data quality has been checked for, the next step is to perform various transformations. The list of transformations available in the workflow browser is described below. Each transformation will produce a new child dataset in the navigator. Also, rows and columns in each of these datasets will be *lassoed* with the rows and columns, respectively, in all the other datasets. Selecting a row/column in one dataset with highlight it in all the other datasets and open views, making it easy to track objects across datasets and

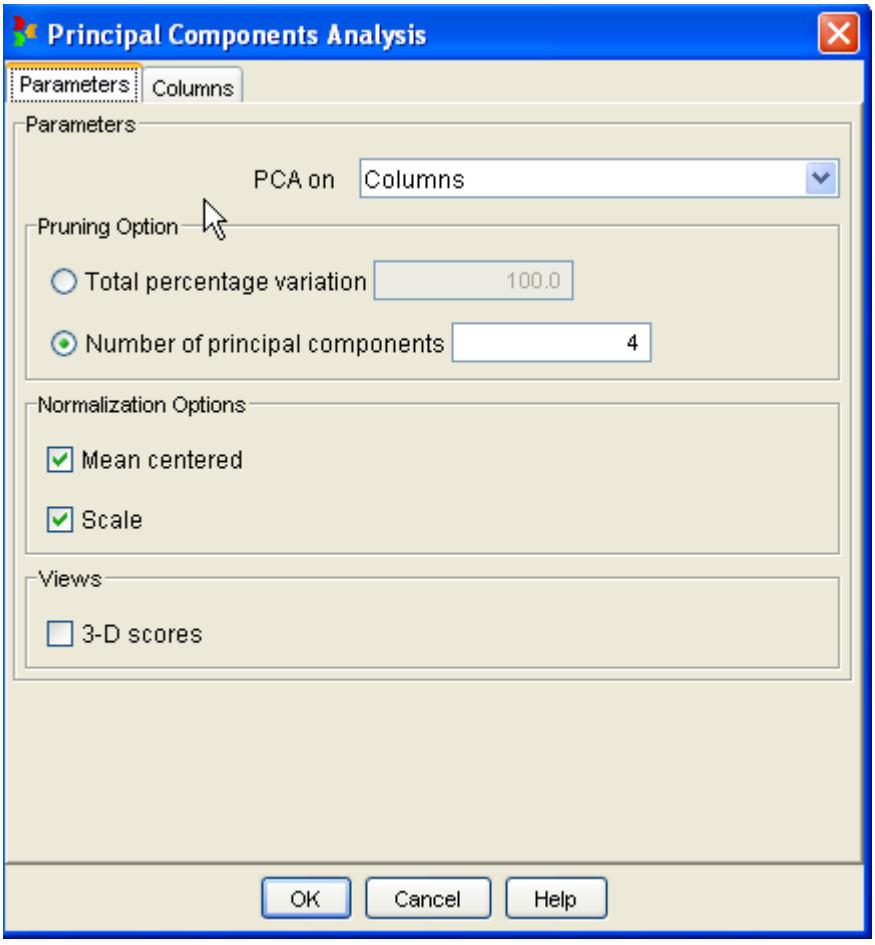
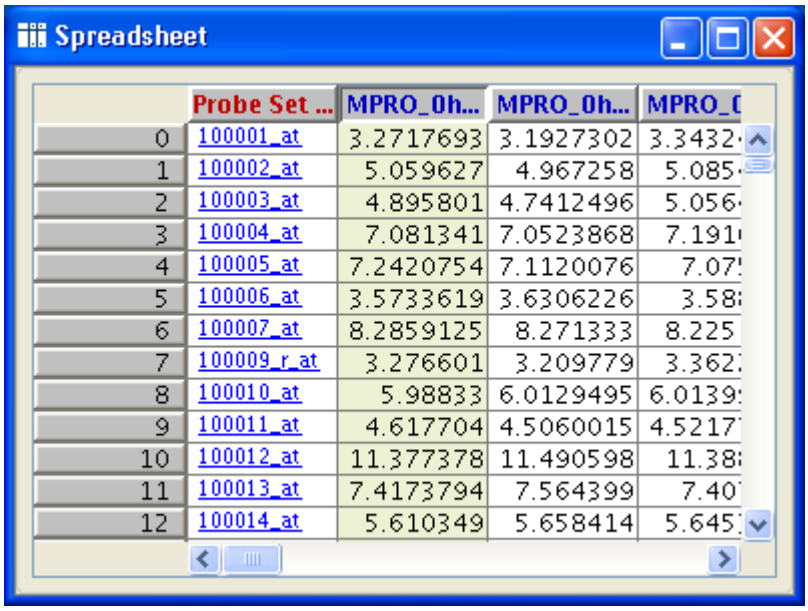


Figure 9.17: PCA



	Probe Set ...	MPRO_0h...	MPRO_0h...	MPRO_(
0	100001_at	3.2717693	3.1927302	3.3432
1	100002_at	5.059627	4.967258	5.085
2	100003_at	4.895801	4.7412496	5.056
3	100004_at	7.081341	7.0523868	7.191
4	100005_at	7.2420754	7.1120076	7.07
5	100006_at	3.5733619	3.6306226	3.58
6	100007_at	8.2859125	8.271333	8.225
7	100009_r_at	3.276601	3.209779	3.362
8	100010_at	5.98833	6.0129495	6.0139
9	100011_at	4.617704	4.5060015	4.5217
10	100012_at	11.377378	11.490598	11.38
11	100013_at	7.4173794	7.564399	7.40
12	100014_at	5.610349	5.658414	5.645

Figure 9.18: New Child Dataset Obtained by Log-Transformation

views.

NOTE: Data transformation will often require you to select a specific dataset in the navigator. For example, Log-Transformation will require selecting a Summarization dataset containing signal values (obtained via one of the summarization algorithms or via the import of CHP files). Appropriate messages will be displayed if the right dataset is not selected in the Navigator.

- Filter on Signals: This link can be used to filter out signal values with low variations. Choose one of the options from the pop up window.
- **Variance Stabilization.** Use this step to add a fixed quantity (16 or 32) to all linear scale signal values. This is often performed to suppress noise at log signal values, e.g., as shown in the pre- and post- variance stabilization scatter plots generated by PLIER summarization. Log transformation should be performed only after variance stabilization.

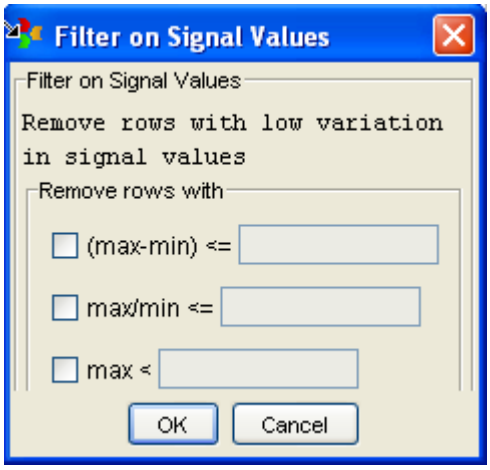


Figure 9.19: Filter on Signals

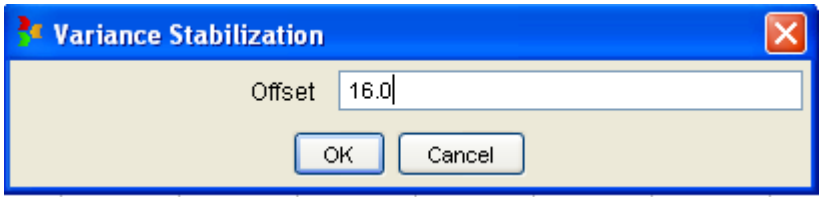


Figure 9.20: Variance Stabilization

- Cy5/Cy3 Ratio: This link takes the ratio of Cy5 signal values with Cy3 signal values for all array.
- **Log Transformation.** Use this step to convert linear scale data to logscale, where logs are taken to base 2. This step is *necessary* before performing statistics, baseline transformations and computing sample averages; these transformations will work only on log-transformed summarized datasets.
- **Baseline Transformation.** This step only works on log-transformed datasets and produces *log-ratios* from log-scale signals. The ratios are taken relative to the average value in a specified experiment group called the *Baseline* group.

  Recall that experiment factors and groups were provided earlier as in Section 5.3.2. One of these groups of replicate arrays will serve as the baseline. Next, the log-scale signal values of each probeset will be averaged over all arrays in the baseline group. This amount will be subtracted from each log-scale signal value for this probeset in the log-transformed summarized dataset. This transform is useful primarily for viewing (e.g., in a heatmap, colors in the baseline group are subdued and all others reflect a color relative to this baseline group, in particular, positive and negative log ratios relative to this group are well differentiated).

  To run this transformation, you will need to specify the baseline group. To this effect, **ArrayAssist** will ask you first to choose an experiment factor amongst those provided prior to generating signal values. Next, it will ask you to choose the baseline group from within the groups for this experiment factor.
- **Compute Sample Averages.** This step only works on log-transformed datasets and averages arrays within the same replicate groups to obtain a new set of averaged arrays. Recall that experiment factors and groups were provided earlier as in Section on The Experiment Grouping. To run this transformation, you will need to specify the experiment factor(s) and group(s) over which averaging needs to be performed. For instance, you may choose one experiment factor and all or a few groups corresponding to this factor; the averages within each of the chosen groups will be computed. If you choose multiple experiment factors, say factor A with groups AX and AY and factor B with groups BX and BY, then averages will be computed within the

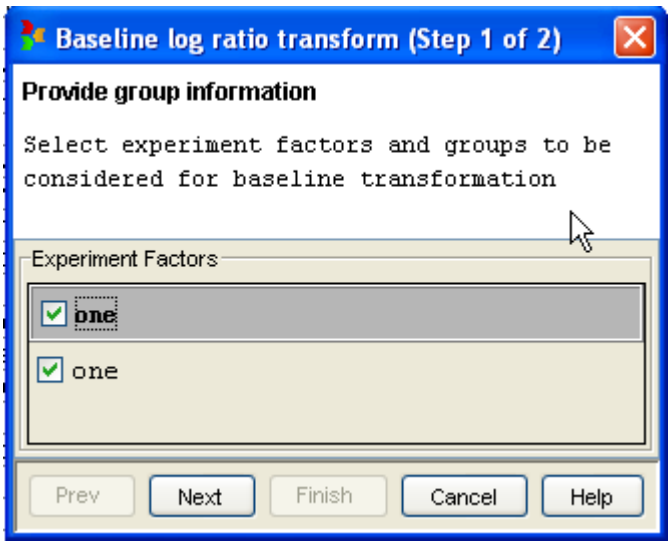


Figure 9.21: Step 1 of Baseline Transformation

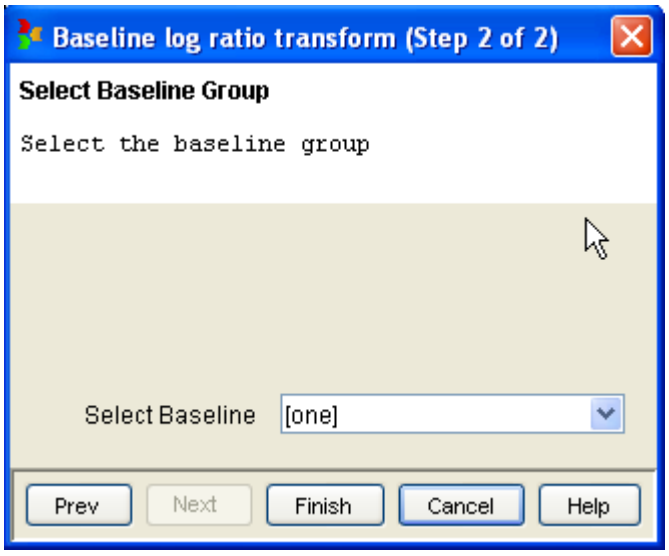


Figure 9.22: Step 2 of Baseline Transformation

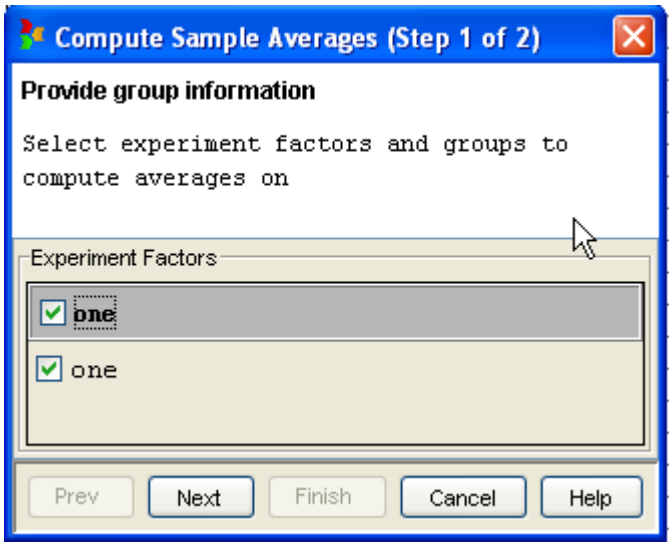


Figure 9.23: Step 1 of Sample Averages

4 groups, AX/BX, AX/BY, AY/BX, and AY/BY. The result of running this transformation will be a new dataset containing the group averages. By using the up/down arrow keys on the dialog shown below, the order of groups in the output dataset can be customized.

- Mean/Median Shift transform: This link shifts each value in the Cy5/Cy3 log ratio column with reference to either the mean or median of that column.
- Dye Swap Transform: This link can be used to mark dye swap data, if applicable. The dye swap pair have to be identified on the pop up window. The second file in each selection is taken as the dye swapped file.
- **Fill In Missing Values.** This step only works on log-transformed datasets and allows missing values in signal columns to be filled in either by a fixed value or via interpolation using the KNN (K Nearest Neighbours) algorithm.
  - Fixed value: All missing values will be replaced by a fixed value. The choice of the fixed value can be entered in the pop up window in 'Replace by' field.
  - KNN Algorithm: The KNN algorithm can be used to fill in

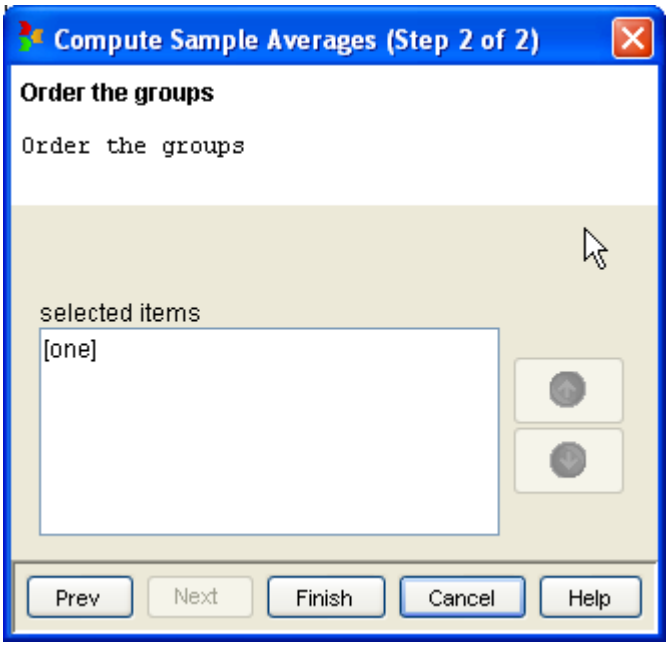


Figure 9.24: Step 2 of Sample Averages

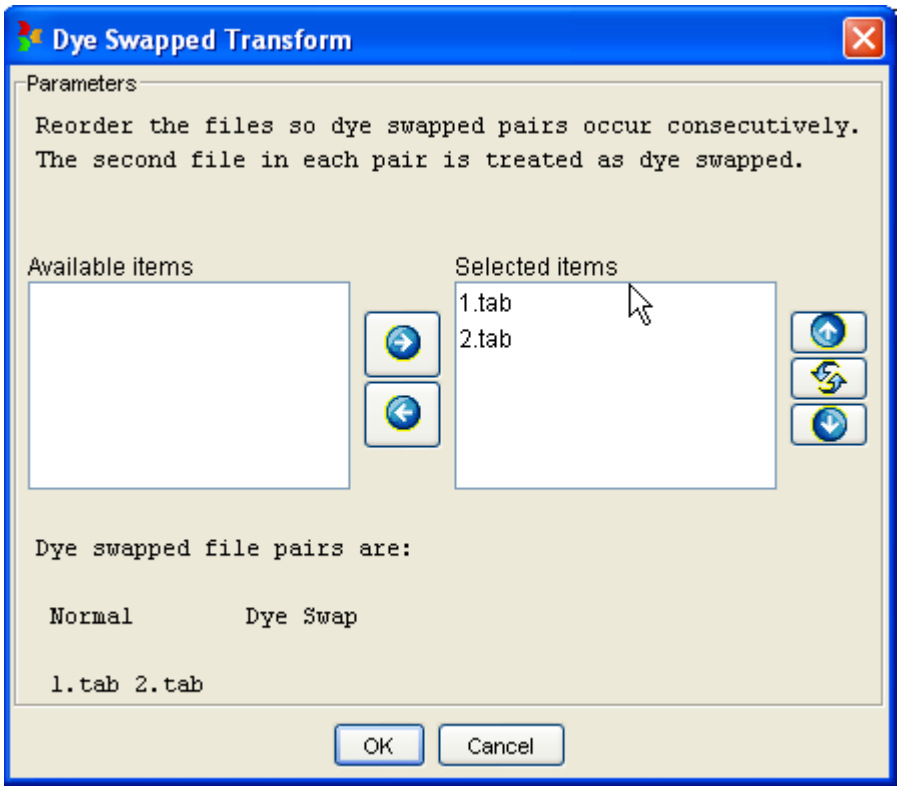


Figure 9.25: Dye Swap Transform

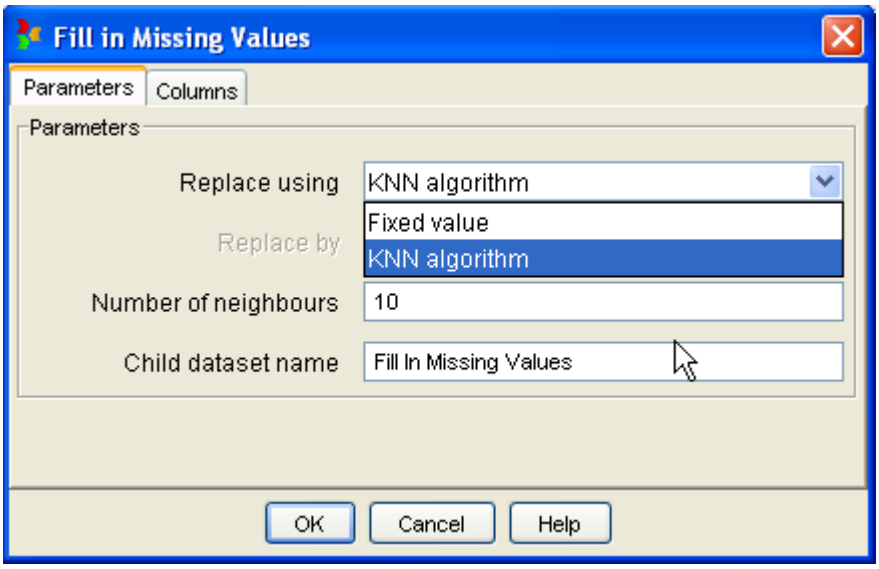


Figure 9.26: Fill in Missing Values

all missing values.

The second tab in the pop up window called *Columns* can be used to pick columns for filling in missing values.

- **Combine Replicate Spots.** This step averages over replicate spots on the arrays. Replicates are identified based on values in a specified column. Note that the averaging works in place, i.e., the average value is repeated for each of the replicate spots rather than reducing each group of replicate spots to one spot each.

9.2.4 Data Viewing

Data in datasets within an Two Dye project can be visualized via the views in the *Views* menu as well as the view icons on the toolbar. Each view allows various customizations via the Right-Click *Properties* menu. Some views which operate on specific columns or subsets of columns will use the column selection in the currently active dataset by default. To select columns in a dataset use Left-Click , Ctrl-Left-Click , Shift-Left-Click on the body of the column (and not on the header). For more details on the various views and their properties, see Data Visualization.

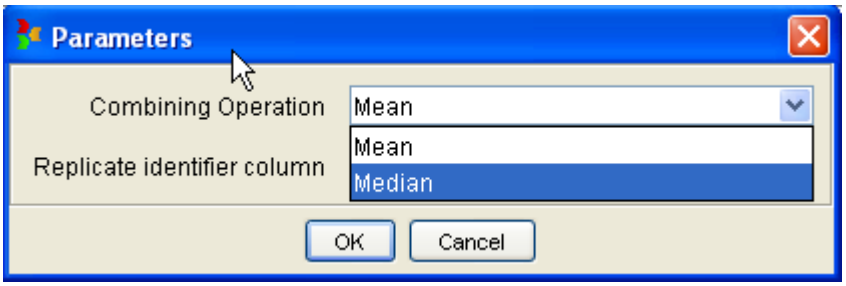


Figure 9.27: Combine Replicate Spots

The Two Dye Workflow browser currently provides the following additional viewing options.

Profile Plot by Groups This view option allows viewing of profiles of probesets across arrays comprising specific experiment factors and groups of interest. Recall that experiment factors and groups were provided earlier as in Section The Experiment Grouping. To obtain this plot, you will need to specify the experiment factor(s) and group(s) over which averaging needs to be performed. For instance, you may choose one experiment factor and all or a few groups corresponding to this factor; you can then also use the up/down arrows to specify the order in which the various groups will appear on the plot. A profile plot with the arrays comprising these groups, in the right order, will be presented.

9.2.5 Significance Analysis

ArrayAssist provides a battery of statistical tests including t-tests, Mann-Whitney Tests, Multi-Way ANOVAs and One-Way Repeated Measures tests. Clicking on the *Significance Analysis Wizard* will launch the full wizard which will guide you through the various testing choices. Details of these choices appear in The Differential Expression Analysis Wizard, along with detailed usage descriptions. For convenience, a few commonly used tests are encapsulated in the Two-Dye Workflow as single click links; these are described below.

NOTE: Significance Analysis requires that Factor and Group information be provided BEFORE signal values are generated. Also the single-click links can only be performed on log-transformed datasets.

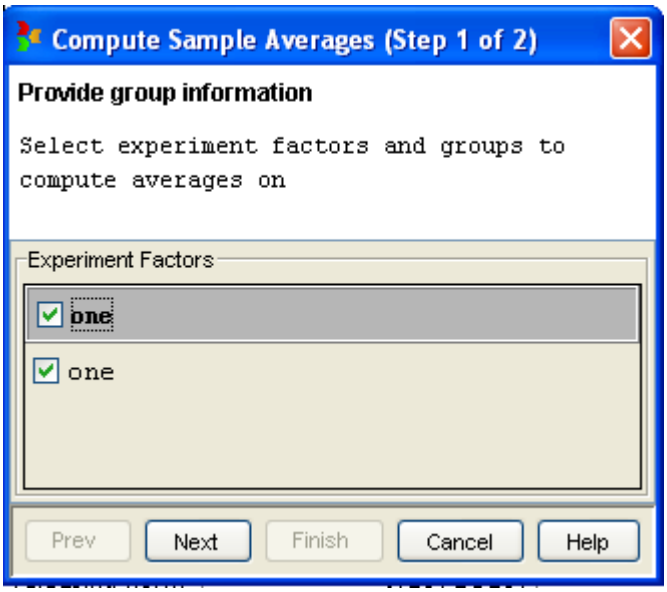


Figure 9.28: Step 1 of Profile Plot by Groups

Treatment vs Control comparison This link will function only if the Experiment Grouping view has only one factor, which comprises two groups. You will be prompted for which of the two groups is to be considered as the Control group. A standard t-test is then performed between Treatment and Control groups. p-values, Fold Changes, Directions of Regulation (up/down), and Group Averages are derived for each probeset in this process. In addition, p-values corrected for multiple testing are also derived using the Benjamini-Hochberg FDR method (see Differential Expression Analysis for details).

Multiple Treatment comparison This link will function only if the Experiment Grouping view has only one factor, which comprises more than two groups. A One-Way ANOVA will be performed on all these groups. p-values and Group Averages are derived for each probeset in this process. In addition, p-values corrected for multiple testing are also derived using the Benjamini-Hochberg FDR method (see Differential Expression Analysis for details).

Significance Analysis Wizard This link invokes the differential expression wizard. This can be used to run any parametric or non-parametric

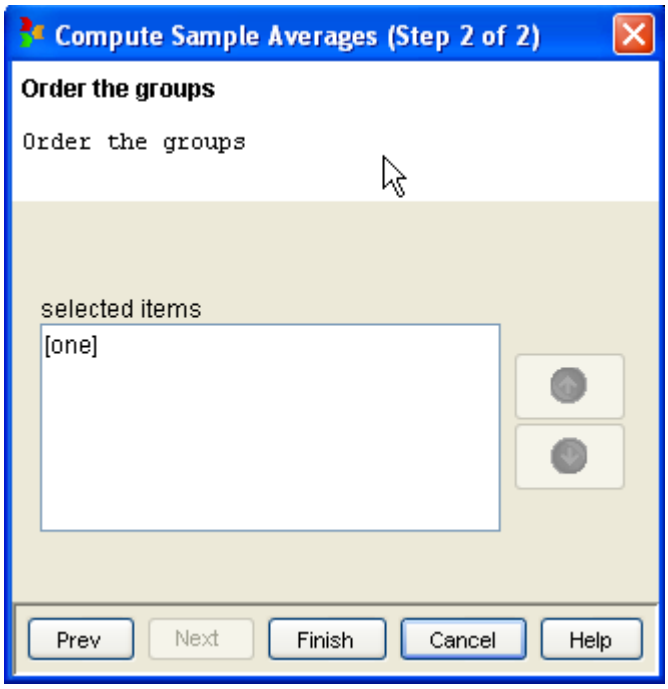


Figure 9.29: Step 2 of Profile Plot by Groups

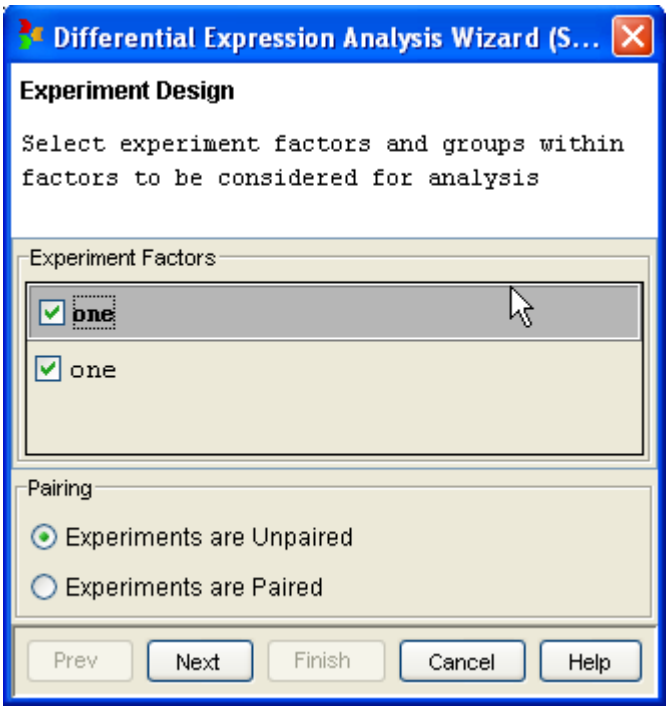


Figure 9.30: Step 1 of Differential Expression Analysis

statistical test along with options for multiple testing correction. Use this option if the experiment set up does not fall into one of the above categories.

Results of Significance Analysis are presented in views and datasets described below. All of these appear under the Diffex node in the navigator as shown below.

The Statistics Output Dataset This dataset contains the p-values and fold-changes (and other auxiliary information), generated by Significance Analysis.

The Differential Expression Analysis Report. This report shows the test type and the method used for multiple testing correction of p-values. In addition, it shows the distribution of genes across p-values and foldchanges in a tabular form. For t-tests, each table cell shows the number of genes which satisfy the corresponding p-value and fold-change cutoffs. For ANOVAs, each table cell shows the number of

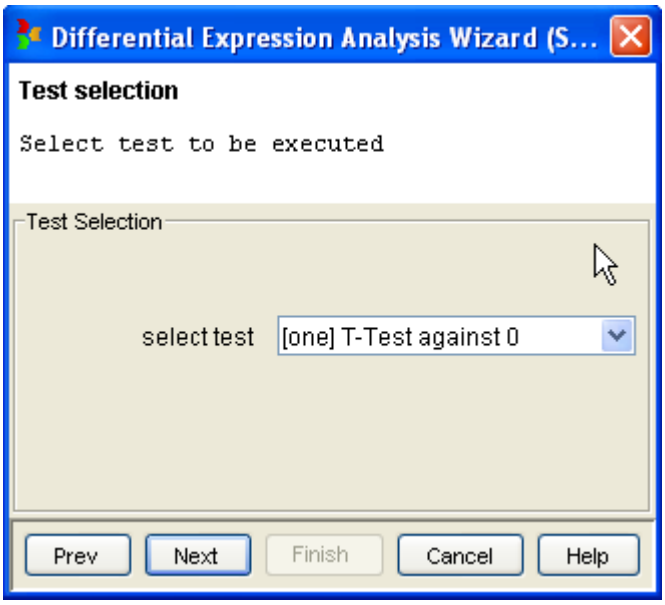


Figure 9.31: Step 2 of Differential Expression Analysis

genes which satisfy the corresponding fold-change cutoff only. For multiple t-tests, the report view will present a drop down box which can be used to pick the appropriate t-test. Clicking on a cell in these tables will select and lasso the corresponding genes in all the views. Finally, note that the last row in the table shows some Expected by Chance numbers. These are the number of genes expected by pure chance at each p-value cut-off. The aim of this feature is to aid in verifying that the number of genes expected by chance is much lower then the actual number of genes found (see Differential Expression Analysis for details).

The Volcano Plot. This plot shows the log of p-value scatter-plot against the log of fold-change. Probesets with large fold-change and low p-value are easily identifiable on this view. The properties of this view can be customized using Right-Click Properties.

Filter on Significance Finally, once significance analysis has been done, the dataset can be filtered to extract genes that are significantly expressed. Click on the link and this will pop-up a dialog to provide the significance value and the fold change criteria. This will create a child

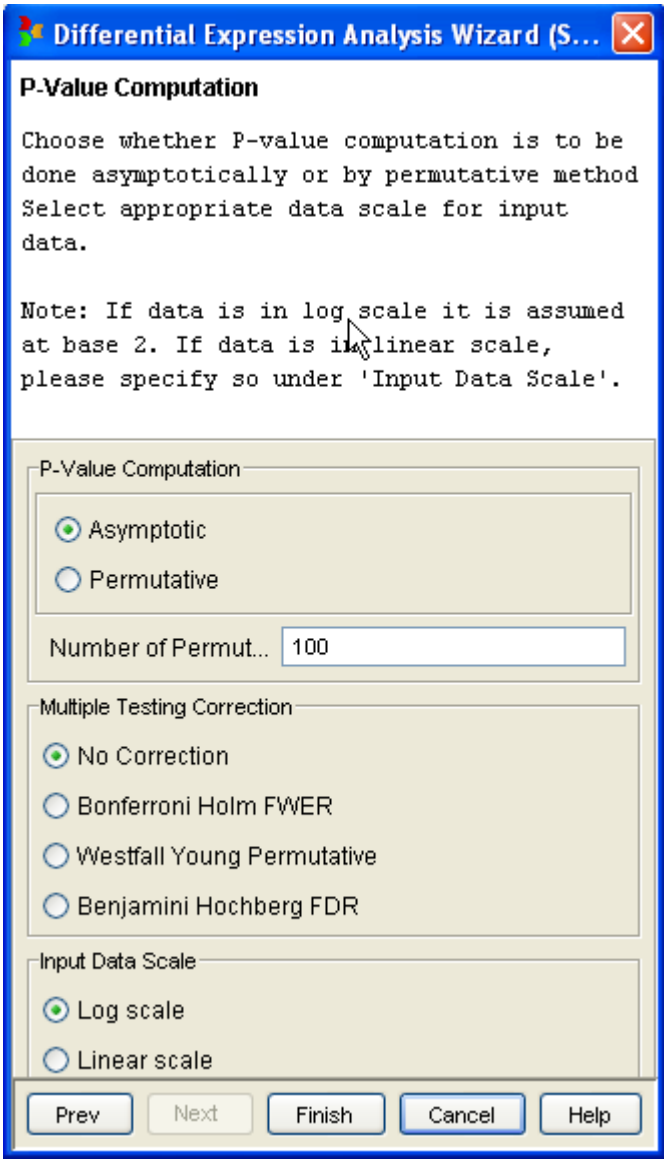
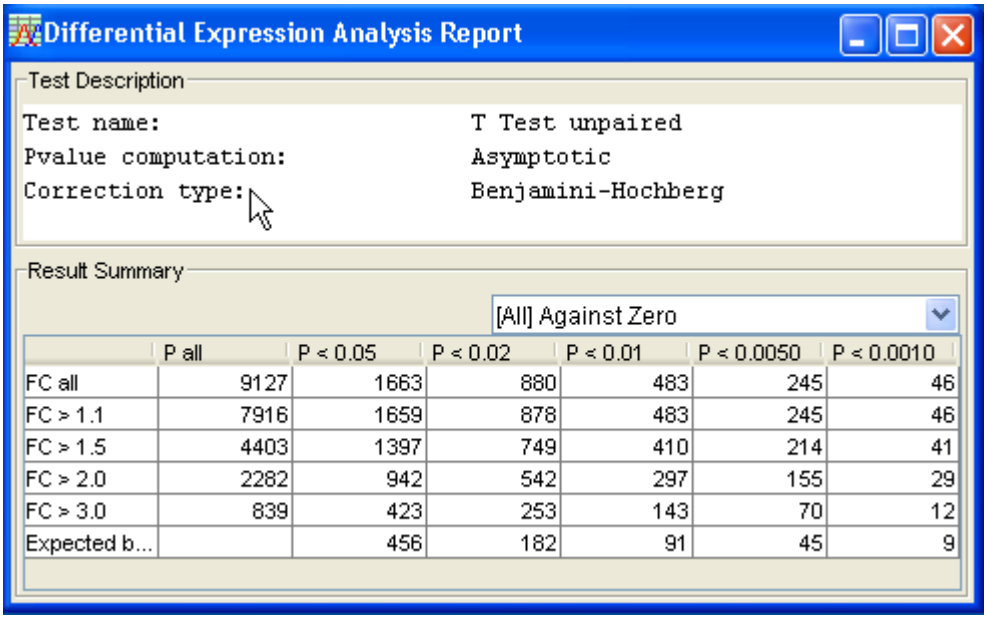

Differential Expression Analysis Wizard (S...
P-Value Computation
Choose whether P-value computation is to be done asymptotically or by permutative method Select appropriate data scale for input data.
Note: If data is in log scale it is assumed at base 2. If data is in linear scale, please specify so under 'Input Data Scale'.
P-Value Computation
Asymptotic
Permutative
Number of Permut...
100
Multiple Testing Correction
No Correction
Bonferroni Holm FWER
Westfall Young Permutative
Benjamini Hochberg FDR
Input Data Scale
Log scale
Linear scale
Prev
Next
Finish
Cancel
Help

Figure 9.32: Step 3 of Differential Expression Analysis



	P all	P < 0.05	P < 0.02	P < 0.01	P < 0.0050	P < 0.0010
FC all	9127	1663	880	483	245	46
FC > 1.1	7916	1659	878	483	245	46
FC > 1.5	4403	1397	749	410	214	41
FC > 2.0	2282	942	542	297	155	29
FC > 3.0	839	423	253	143	70	12
Expected b...		456	182	91	45	9

Figure 9.33: Differential Expression Report

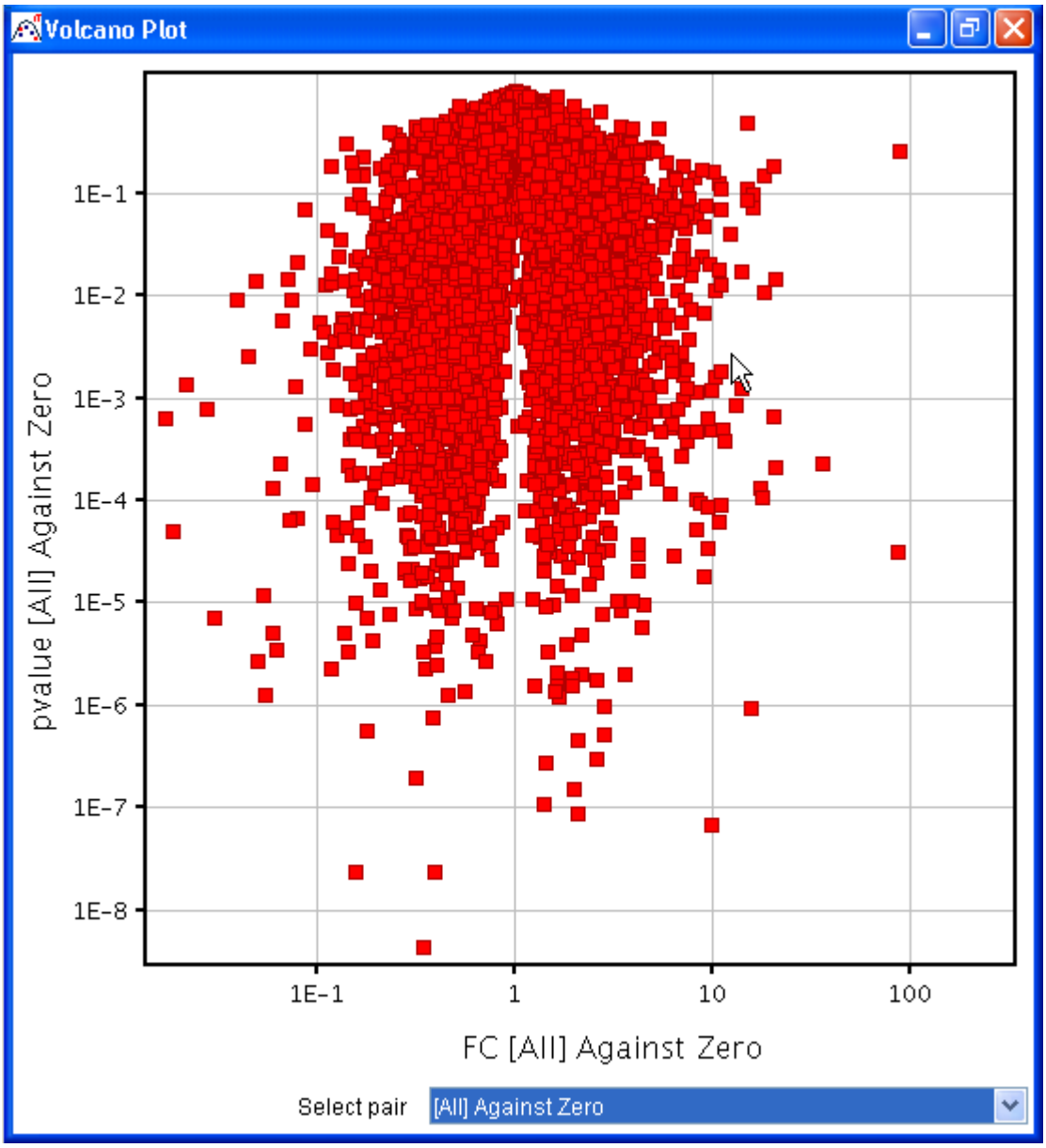


Figure 9.34: Volcano Plot

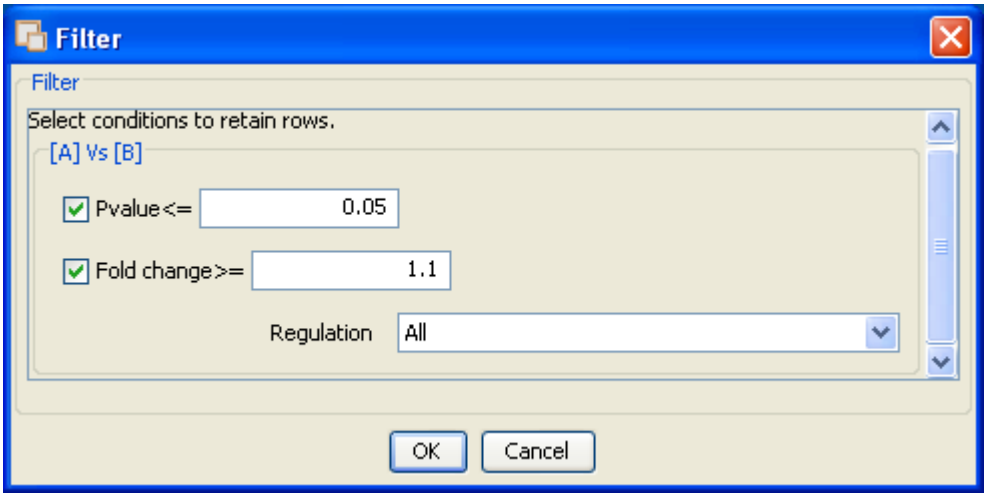


Figure 9.35: Filter on Significance Dialog

dataset with the set of genes that satisfy the filter critera provided.

9.2.6 Clustering

The only clustering link available from the workflow browser is the K-Means which clusters the signal columns into 10 clusters. To run another algorithm or to change parameters, use the *Cluster* menu. See Section Clustering for more information.

9.2.7 Save Probeset List

Create Probeset List from Selection This link will create a probeset or Gene List from the selected genes. Normally, after identifying significantly expressed, you would like to save these genes or probesets of interest in the **ArrayAssist**. This will will save the selected probesets of genes as a gene list that will be available in any place in the tool. You will have to provide a name for the probeset or gene list and the mark to be used to associate with the list.

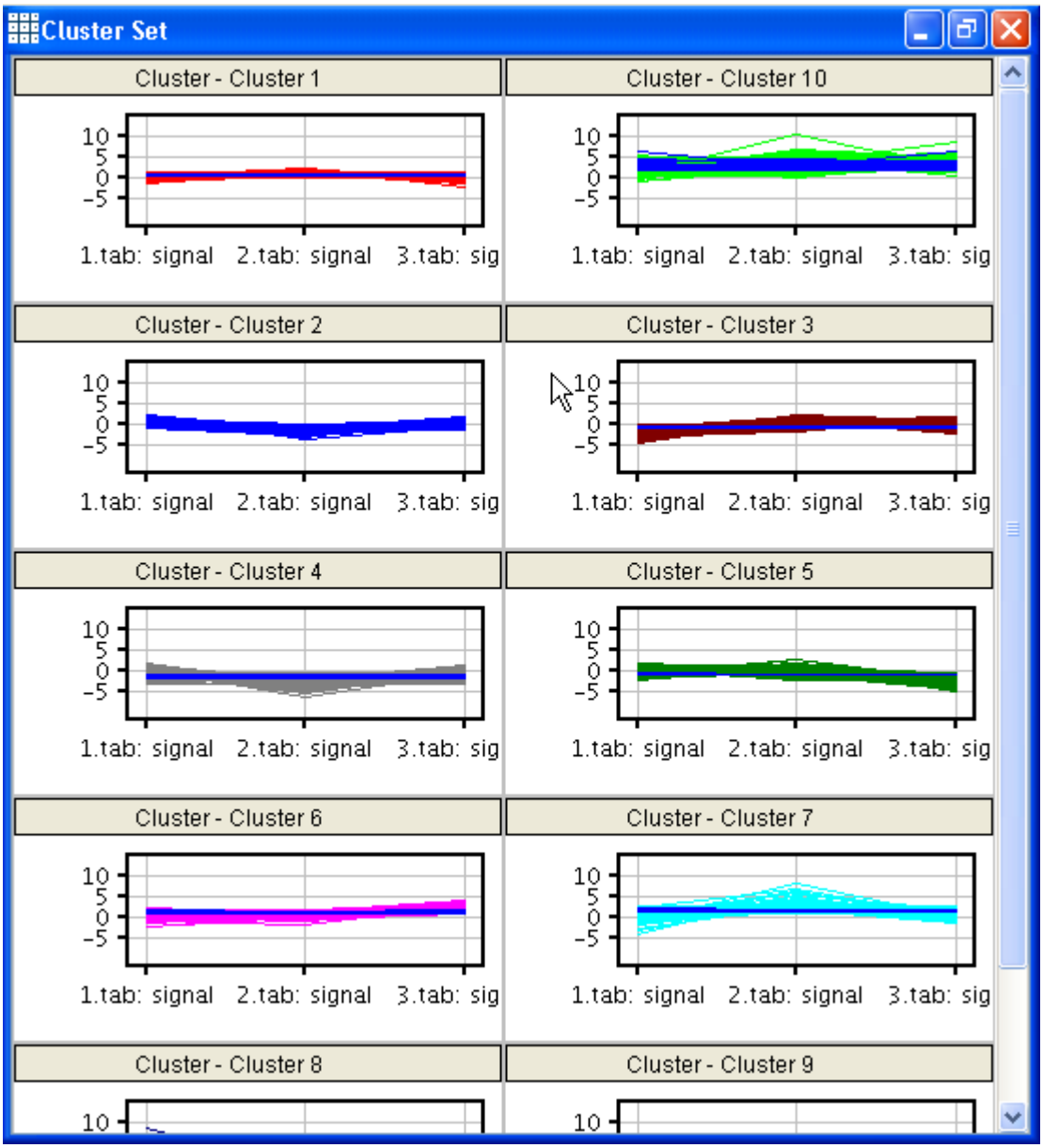


Figure 9.36: K-means Clustering

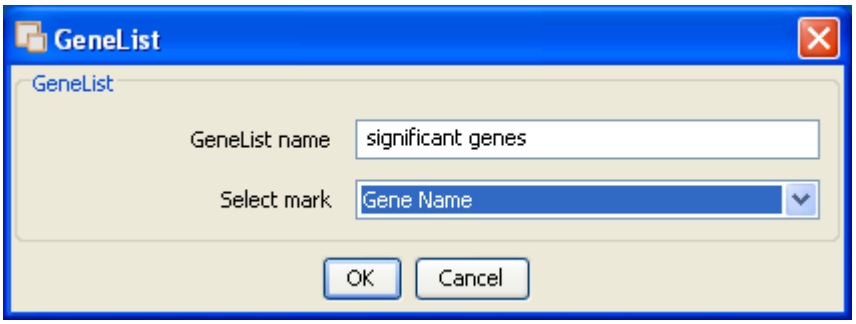


Figure 9.37: Create Probeset List from Selection

9.2.8 Import Gene Annotations

Once significant genes have been identified, you may want to explore the biology of the genes by bringing in annotations of the genes from a file, or annotating genes from various web sources via the annotation engine in **ArrayAssist**. The following links allow you to import and fetch annotations into the dataset.

Import Gene Annotations from File If you have your own set of gene annotations which you wish to import, prepare these annotations as a tab or comma separated file with genes as rows and annotation fields (name, symbol, locuslink etc.) as columns. Then import this file by going to the gene annotations dataset and using *Data* $\longrightarrow$ *Columns* $\longrightarrow$ *Import Columns*. Provide the file name and the gene identifier to be used for synchronizing columns in the file imported with columns in the gene annotations dataset. Next, mark each of the imported columns by setting the appropriate column mark in the *Data Properties* (appropriate marks include Unigene Id, Gene Name etc.). This will ensure two things: first, that these new columns are available from all child datasets, and second, that these columns are interpreted correctly by the annotation modules (web spidering, GO Browsing etc).

Mark Annotation Columns This link can be used to mark columns, i.e., identify as Unigene, Genbank Accession etc. Alternatively, to mark a column, use Data →Data Properties and set the appropriate marks using the dropdown list provided for each column.

Fetch Gene Annotations from Web You can fetch annotations for selected genes from various public web sources. Select the genes of inter-

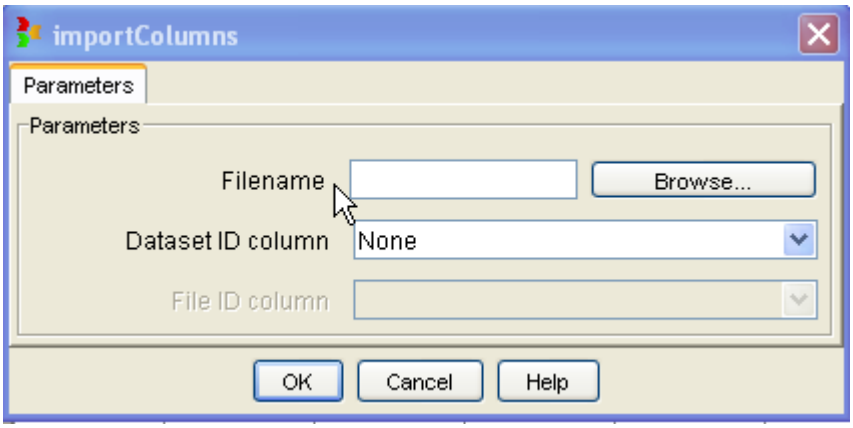


Figure 9.38: Import File

est from any dataset or view, then choose the gene annotations dataset on the Navigator and click on this link. Select the public source of your interest, and indicate the input gene identifier you wish to start with (Unigene, Genbank Accession etc) and the information you need to fetch (gene name, alias etc). The information fetched will be updated in the gene annotations dataset or appended in some cases when the column fetched is not already there in the dataset. Note that the input identifiers used need to be marked (see Section Marking Annotation Columns), i.e., identified as Unigene, Genbank Accession etc. To mark a column, use *Data* $\longrightarrow$*Data Properties* and set the appropriate marks using the dropdown list provided for each column. Alternatively, the Annotation wizard has an option to mark columns. For more details on the public sites accessible and of the input and output identifiers, see Section Annotating Genes.

• Note that several marked gene annotation columns are hyperlinked, for instance the Probeset Id is linked to the Affymetrix NetAffx page, Gene Ontology accession is linked to the AMIGO page etc. For a list of these hyperlinks, see *File*$\longrightarrow$*Configuration*$\longrightarrow$*AffyURL*. These hyperlinks can be edited here.

9.2.9 Discovery Steps

This section contains links to dicover the biology of the selected genes by examining the GO terms associated with the selected genes or to visualize

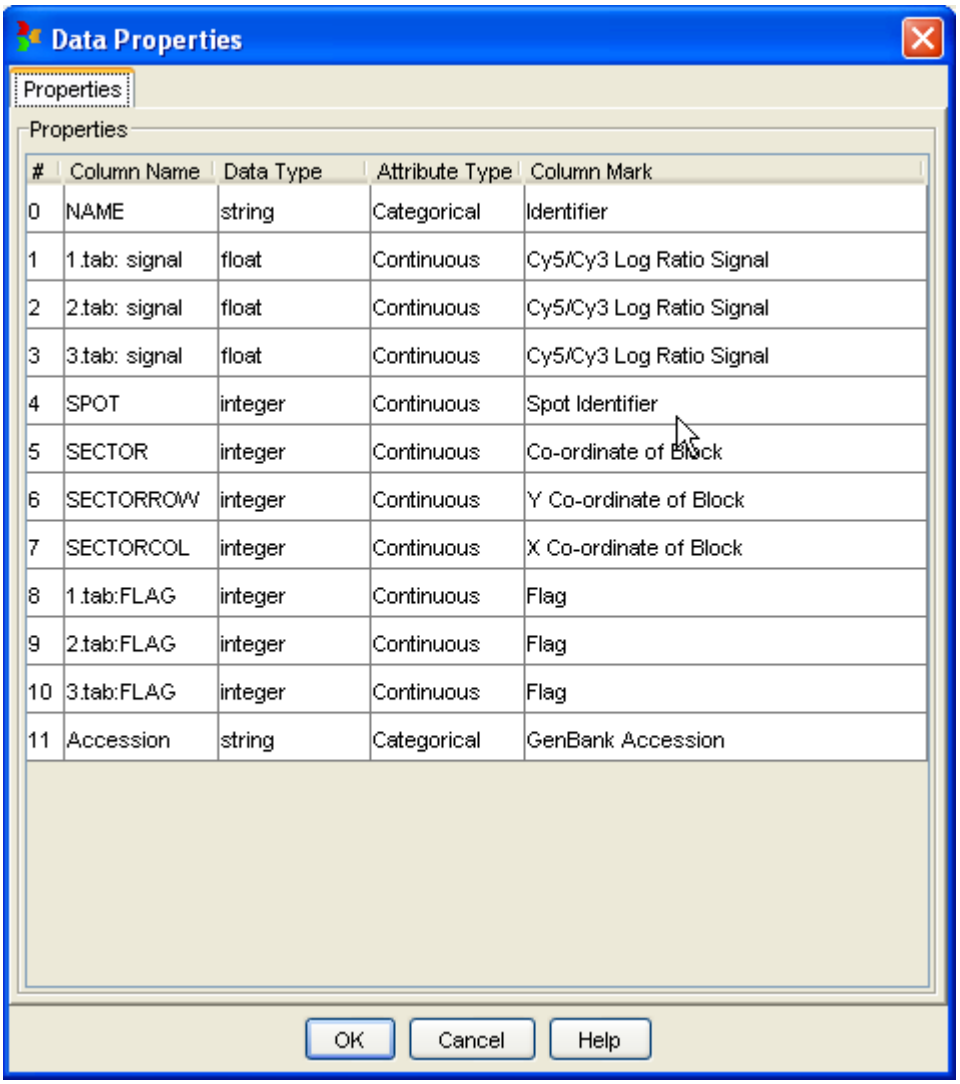

Data Properties
Properties
Properties
Column Name Data Type Attribute Type Column Mark
0 NAME string Categorical Identifier
1 1.tab: signal float Continuous Cy5/Cy3 Log Ratio Signal
2 2.tab: signal float Continuous Cy5/Cy3 Log Ratio Signal
3 3.tab: signal float Continuous Cy5/Cy3 Log Ratio Signal
4 SPOT integer Continuous Spot Identifier
5 SECTOR integer Continuous Co-ordinate of Block
6 SECTORROW integer Continuous Y Co-ordinate of Block
7 SECTORCOL integer Continuous X Co-ordinate of Block
8 1.tab:FLAG integer Continuous Flag
9 2.tab:FLAG integer Continuous Flag
10 3.tab:FLAG integer Continuous Flag
11 Accession string Categorical GenBank Accession
OK
Cancel
Help

Figure 9.39: Mark Annotation Columns

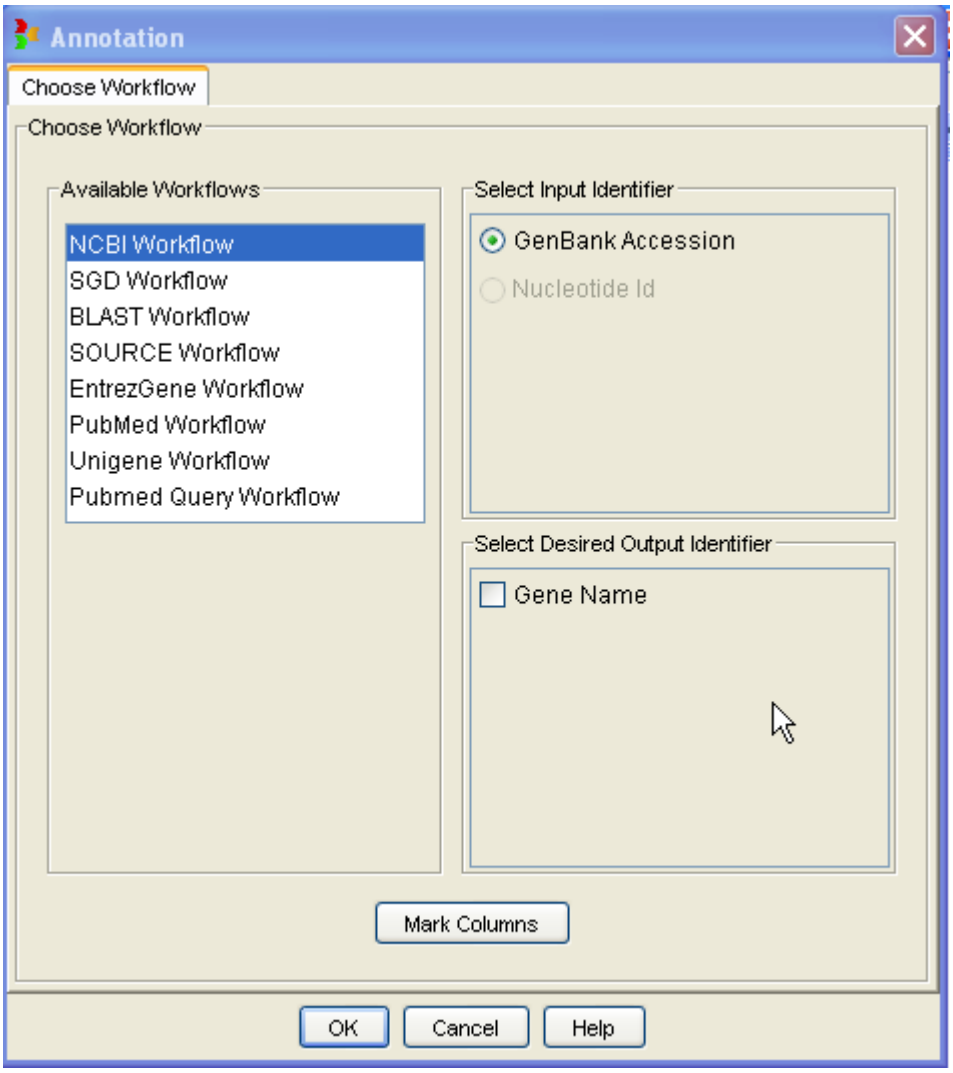


Figure 9.40: Fetch Gene Annotations

the location of the selected genes on the Chromosome viewer, if the gene location information is available in the dataset.

GO Browser You can view *Gene Ontology* terms for the genes of interest in the Gene Ontology Browser invokable from this link. This browser offers several queries, a few of which are detailed below. See Section GO Browser for a more complete description.

NOTE: To launch the GO browser, your currently active dataset needs to contain a Gene Ontology Accession column and this must be marked as such a column via *Data* $\longrightarrow$*Properties*. Each cell in this column should be a pipe separated list of GO terms, e.g., GO:0006118|GO:0005783|GO:0005792|GO:0016020.

- To view GO Terms for genes of interest and to identify enriched GO Terms, select genes of interest from any view and then click on the Find GO Terms with Significance  icon.

  Next move to the *Matched Tree view*. Here you will see all Gene Ontology terms associated with at least one of the genes along with their associated enrichment p-value (see Section GO Computation for details on how this is computed). You can navigate through this tree to identify GO Terms of interest.

- A tabular view of the p-values can also be obtained by clicking on the p-value Dataset  icon. This will produce a table in which rows are the above visible GO terms, and the columns contain various statistics (i.e., enrichment p-value, the number of genes having a particular GO term in the entire array, the number of genes amongst those selected having a particular GO term etc.).

- Another tabular dataset can be obtained by clicking on the Gene Vs GO Dataset  icon and providing a cut-off p-value. This dataset shows probesets along the rows and GO Terms which occur in at least one of these probesets along the columns, with each cell being 0 or 1 indicating the presence or absence of that GO term for that probeset. This view is best viewed as a HeatMap by selecting the relevant columns and launching the HeatMap view from the *View* menu.

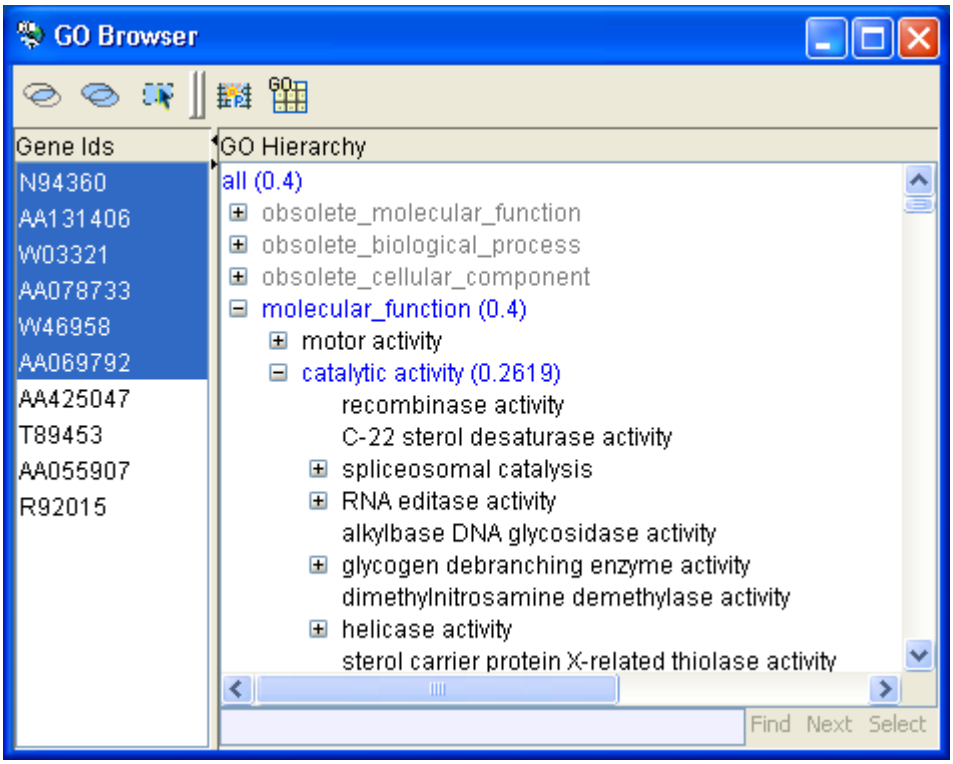


Figure 9.41: GO Browser

- You can also begin with a GO term (select it in the Full Hierarchy tab, if necessary you can use the search function to locate the term), and then click on Find All Genes with this Term  icon. This will select all probesets having this particular GO term in all the views and datasets.

Viewing Chromosomal Locations. Click on this link to view a scatter plot between Chromosome Number and Chromosome Start Location. Each probeset is depicted by a thin vertical line. Each chromosome is represented by a horizontal bar. Each probeset can be given a color as well. For instance, to color probesets by their fold changes or p-values, go to the Statistics output dataset in the Navigator and then launch the Chromosome Viewer. Use Right-Click *Properties* to color by the p-value or fold change columns.

NOTE: To launch the chromosome viewer, your currently active dataset needs to contain a Chromosome start location column and a Chromosome number column and this must be marked as such via *Data* $\longrightarrow$*Properties*.

Creating Custom Links. You can cause entries in a particular column to be treated as hyperlinks by changing the column mark to URL in *Data* $\longrightarrow$*Data Properties*. Subsequently, clicking on an entry in this column (either in the spreadsheet or in the lasso) will open the corresponding link in an external browser. Note that the entries in this column must be hyperlinks (i.e., of the form http:// etc.).

In case you wish to create a new hyperlink column, use the *Data*$\longrightarrow$*Column* $\longrightarrow$*Append Columns By Formula* command to create an appropriate string column and then use *Data* $\longrightarrow$*Data Properties* to mark this column as a URL column. For more details on creating new columns with formulae, see Section GO Computation.

9.2.10 Genome Browser

Genome Browser The Genome Browser can be invoked using this link. This browser allows viewing of several static prepackaged tracks. In addition, new tracks can be created based on currently open datasets. For more details on usage, see Section The Genome Browser.

Chapter 10

Annotating Results

ArrayAssist provides mechanisms or workflows for automatically retrieving gene information from web sources and viewing this information. All of these workflows are accessible from the Annotation menu in **ArrayAssist**. The annotation module also has other valuable tools which can help relate expression data to biological information, in particular the Gene Ontology (GO) Browser and the GO enrichment values, a basic chromosome viewer, etc.

A normal workflow would be to complete numerical analysis of data distilling a few genes that are significant. The biological information on these genes is then retrived from various sources on the internet directly from **ArrayAssist**. To retrive information from the web, the dataset needs to contain certain columns that are marked as gene identifiers. The **ArrayAssist** then uses these gene identifiers, runs a choosen workflow depending upon the available gene identifies, spiders the web, querries various web sites, retrives information about these genes from the web site, and presents them to the use in **ArrayAssist**. With new information retrived from web sources, more workflows can be run retriving more information. **ArrayAssist** also has certin tools to analyse the retrived information, like enrichment analysis of GO terms in the selected genes, creating a GO dataset for further analysis, etc. The Annotation module thus provides an integrated functionality to access the 'state of the art' information on the genes of interest and infer and interpret the biological role and significance of selected genes in the dataset.

The annotation process follows the steps given below:

1. Import annotation columns into the current dataset.

2. Mark the annotation columns in the dataset from the Data properties and assign appropriet marks to the columns that contain some annotation information. You should have atleast one annotation column in the dataset to start the annotation workflow. Marking annotation columns in the dataset is an essential step to running annotation workflows.

3. Choose and configure a workflow from among the alternatives available. The available workflows depends upon the annotation columns that are marked in the dataset.

4. Retrieve annotation information. This is described in the following section on Annotation Genes from the Web

5. Use the GO Browser and GO Clustering features to explore relationship between data and function.

6. Construct comprehensive PubMed queries for genes using automatically downloaded aliases and symbols, Results are retrieved from PubMed using this query.

7. Analyse the biological significance and biological role of the selected genes from the annotated information.

10.1 Configuration

All the columns in the dataset that are marked as annotation columns are hyperlinked to an appropriate web site. Thus Left-Click in the cell of any annotation column will open a browser with the appropriate page. The URL link for each marked column is set in the configuration of the **ArrayAssist** and can be changed from the configuration or options dialog. Any changes in the configurations or options dialog are effective immediately.

Gene Features or Web Shortcuts

All columns in the Annotation Table except PubMed Id column are hyperlinked to point to a webpage containing information about that columns. Thus the information in each cell of the Annotation Table is hyperlinked to fetch information from the web. These hyperlinks can be modified to point to a webpage different from the default in **ArrayAssist**. The term **%arg1** is replaced by the element in the cell to create the URL string. Eg.

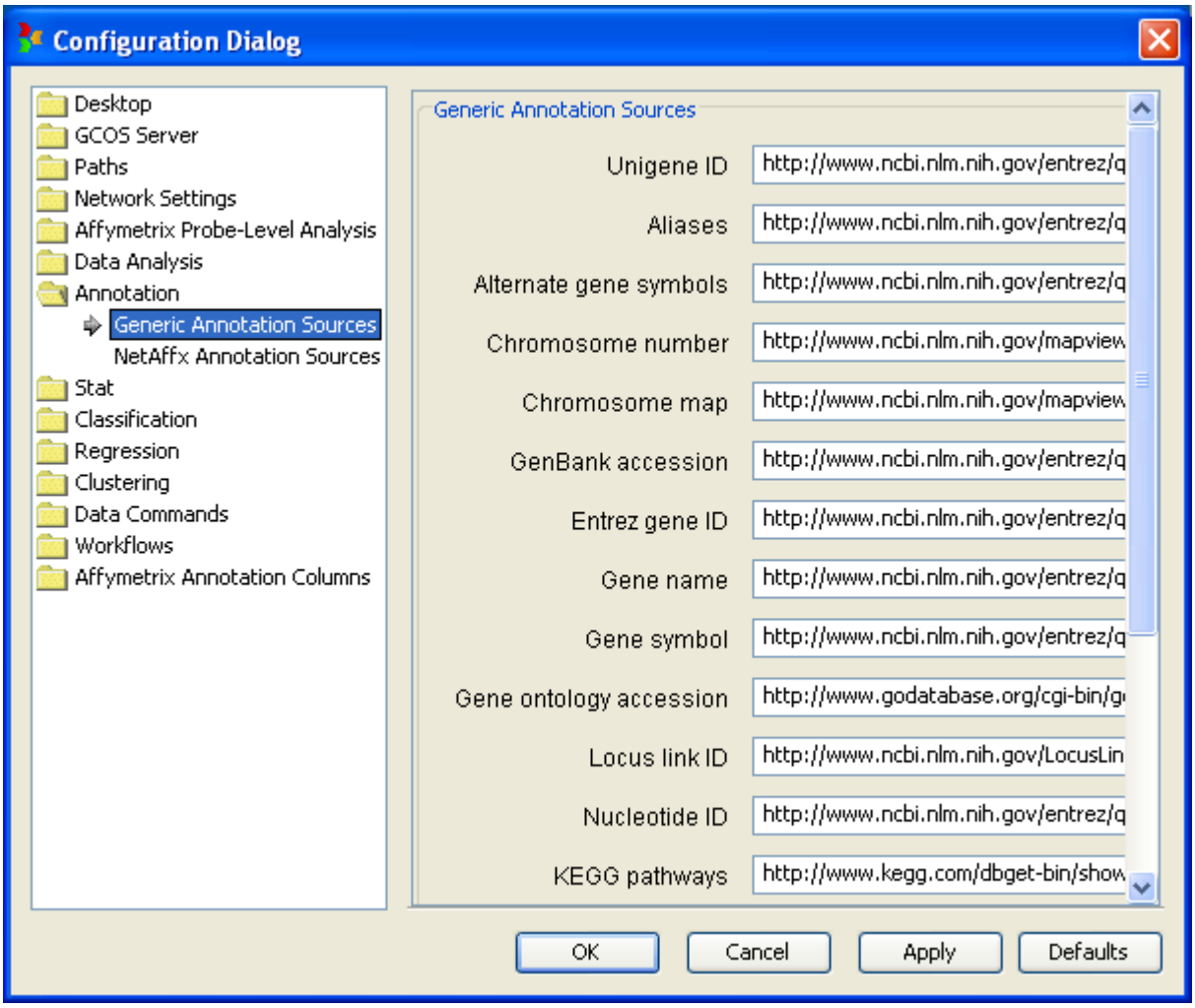


Figure 10.1: Configuring Annotation Database

If the user clicks on a cell containing a UniGene ID **Hs.73875** and the web shortcut for UniGene has been set to http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Search\&db=unigene\&term=%arg1 in the configuration, the web link would point to http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Search\&db=unigene\&term=Hs.73875. The default URLs for the marked annotation columns are available in *Tools* $\longrightarrow$ *Options*.

10.2 Annotation Genes from the Web

To start the annotation process, the dataset must contain gene identifies recognized by various public databases and internet sites, like the Unigene Id, Locus Link Id, Entrez gene Id, etc. Further the columns that contain such gene identifiers must be marked at an annotation column with the appropriate mark, so that the **ArrayAssist** can indentify such columns and use the information the the column to access data from various web sorces.

10.2.1 Marking Annotation Columns

The first step in the annotation process is to identify and mark columns in the dataset that would be used in the annotation process. Columns in the dataset are marked with appropriate annotation marks from the data properties dialog. The data properties dialog shows all the columns of the dataset; the data type and attribute type of each of the columns; and the column marks if any for each column. To mark a column in the dataset as an annotation column, identify the appropriate column in the dataset. In the **Column Marks** column of the data properties dialog, choose the correct mapping column from the Drop-Down-List . All annotation marks in the Drop-Down-List will be colored with the same color. Also the column headers of all columns in the dataset that have been given annotation marks will be shown a unique color.

The list below gives the annotation marks currently available in **ArrayAssist**.

- Unigene Id
- Aliases
- Alternate gene symbols
- Chromosome Number

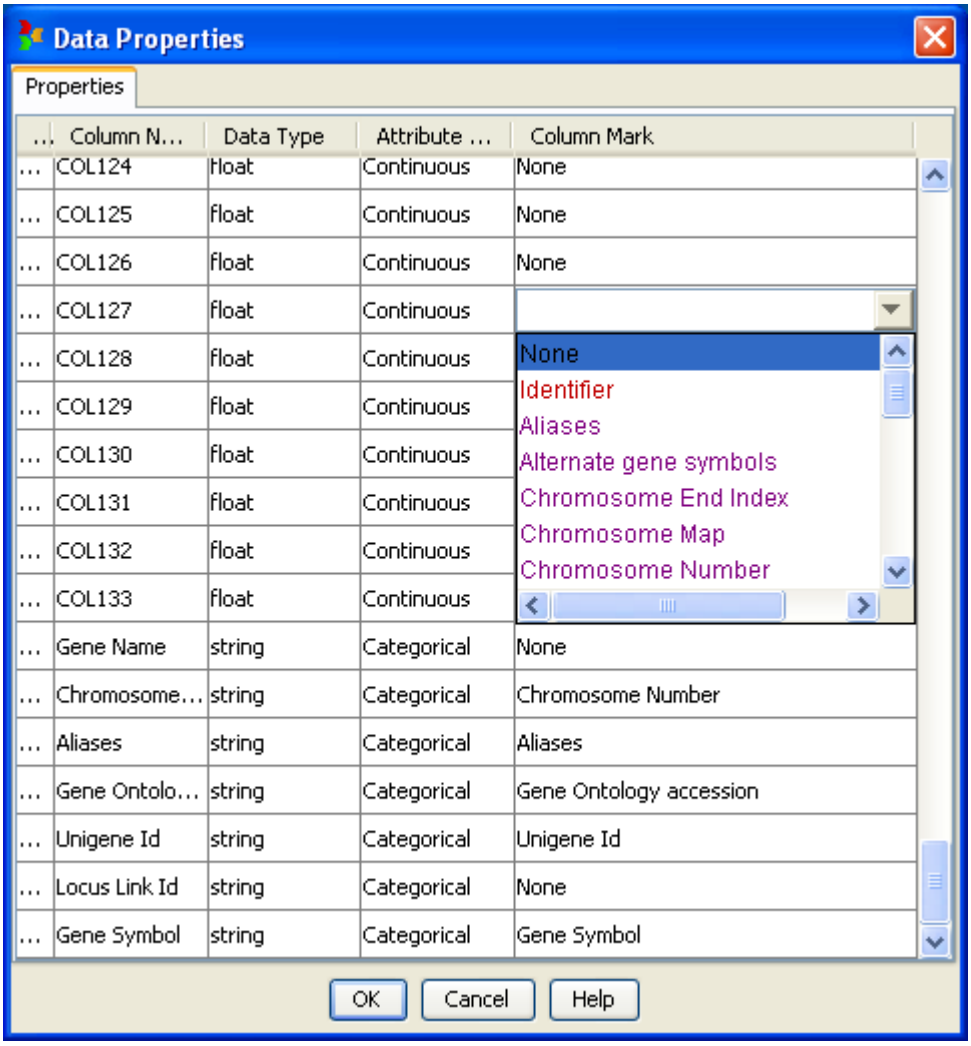


Figure 10.2: Mapping Annotation Identifiers

- Chromosome Map
- GenBank Accession
- Entrez Gene Id
- Gene Name
- Gene Symbol
- Gene Ontology accession
- Locus Link Id
- Nucleotide Id
- KEGG Pathways
- Pubmed Query
- Pubmed Ids
- SGD Id
- GenBank Accession Retrieved After Blast
- Standard Name of yeast gene
- Systematic Name of yeast gene
- Chromosome Start Index
- Chromosome End Index

10.2.2 Starting Annotation

To start the Annotation process, launch the annotation dialog from the menu bar or from the appropriate workflow link in the workflow browser. A few genes or rows of the dataset must be selected to start annotating from the web. If no genes or rows of the dataset are selected, you will be prompted with an error and resolution dialog asking you to select rows for annotation. If there are rows selected in the dataset, the annotation dialog will be launched. This has three panels. The left panel shows the available workflows, the top right panel shows the input identifies to be selected and the bottem right panel shows the set of output identifiers.

Depending upon the workflow and the marked annotation columns in the dataset, the appropriate options in the right panel will be enabled.

If there are no annotation marks in the dataset, none of the workflows will be abailable. The **Mark Columns** button at the bottom of the annotation dialog will launch the data properties dialog enabling you to mark appropriate annotation columns of the dataset. For details on the available marks and the to mark annotation columns refer the section Marking Annotation Columns above.

10.2.3 Running an Annotation Workflow

ArrayAssist provides the ability to annotate genes from the web. **ArrayAssist** has workflows that will visit one or more websites and gather information about a selected gene. The workflow can be used to annotate a gene for the first time or for updating annotation information. The workflows available are described below and required input and output fields for the workflow are listed in ArrayAssist Workflows. Workflows will run only on selected genes.

SOURCE Workflow: A batch query is submitted to Stanford SOURCE site and information is retrieved and used to populate the Annotation Table. This flow is available only for *Homo sapiens*, *Mus musculus* and *Rattus norvegicus* (as of July 25, 2003). Information retrieval is very fast compared to other flows.

Entrez Gene Workflow: The gene id is submitted to Entrez Gene database and all available information for that gene is retrieved.

UniGene Workflow: The gene id is submitted to UniGene and available information for the gene is fetched.

NCBI Workflow: The Gene Name is fetched from NCBI-Nucleotide database.

BLAST Workflow: A BLAST is performed at NCBI. The GenBank Accession number of the first non-clone with lowest e value < 1 is selected.

PubMed Query Workflow: A query string is derived by concatenating user-defined combinations of Aliases, Symbols, Alternate Gene Symbols and Gene Names for a gene with the "OR" condition. String containing the word "EST" are excluded. If the available material is less than 2 characters long no query string is created. The Standard Name, Alias and Systematic Name are used to construct the PubMed

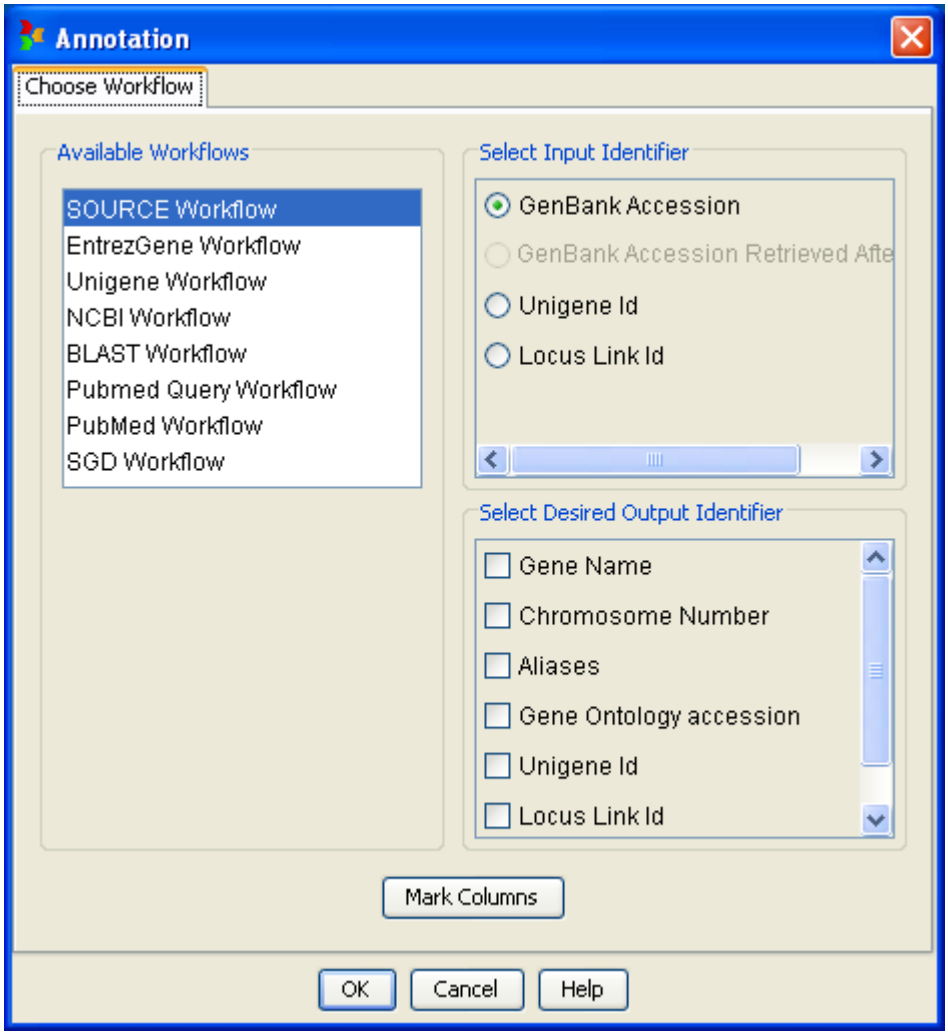


Figure 10.3: Annotation Dialog

query string for yeast genes. This Workflow would be run prior to running a PubMed Workflow. The generated query strings are editable. The PubMed Query can be edited in the Editor window on top of the Annotation Table.

PubMed Workflow: The PubMed query for selected genes are submitted to PubMed and the results retrieved. The PubMed Ids are stored to a temporary file and if desired, the would need to be saved independently. The total number of hits for each gene from teh query is appended as a coulmn in the dataset.

Note: The PubMed Ids are not saved into the session.

SGD Workflow: This flow is applicable only for Yeast genes/Ids. The gene id is submitted to the Saccharomyces Genome Database and all available information is retrieved from SGD. If there are multiple hits, the first one is retrieved.

The table below provides an overview of the different workflows available in **ArrayAssist** along with the inputs and the outputs for each workflow.

10.3 Exploring Results

10.3.1 Working with Gene Ontology Terms

The Gene OntologyTM (GO) Consortium maintains a database of controlled vocabularies for the description of molecular functions, biological processes and cellular components of gene products. The GO terms are represented as a Directed Acyclic Graph (DAG) structure. Detailed documentation for the GO is available at the Gene Ontology homepage (http://geneontology.org). Other databases such as LocusLink and SGD utilize GO terms to describe the gene products in their repertoire and this information is retrieved by **ArrayAssist**. It is displayed in the Gene Ontology column with associated Gene Ontology Accession numbers. A gene product can have one or more molecular functions, be used in one or more biological processes, and may be associated with one or more cellular components. Each GO term is derived from one or more parent terms.

The GO browser can be invoked only if gene ontology information is available for genes in the annotation view.

Workflow	Input	Outputs
SOURCE	Genbank Accession, UniGene Id, LocusLink Id	Gene Name, Chromosome Number, Alias, Gene Ontology, UniGene Ids, LocusLink Id, Gene Symbol
EntrezGene	Entrez Gene Id, LocusLink Id	Gene Name, Chromosome Number, Alias, Gene Ontology, Chromosome Map, KEGG Pathways, UniGene Id, Gene Symbol
UniGene	Genbank Accession, UniGene Id, Nucleotide Id	Chromosome Number, UniGene Id, LocusLink Id
PubMed Query	Gene Name, Gene symbol, Alias, Alternate symbols, Standard Name(Yeast), Systematic Name(Yeast)	Query String for PubMed
PubMed	PubMed Query String	PubMed Ids
BLAST	Genbank Accession	Genbank Accession
NCBI	Genbank Accession Nucleotide Ids	Gene Name
SGD	SGD Ids, Standard Name (Yeast), Systematic Name (Yeast)	Standard Name (Yeast), Gene Ontology, Aliases, Chromosome Number, Systematic Name (Yeast), SGD Id

Table 10.1: **ArrayAssist** Workflows

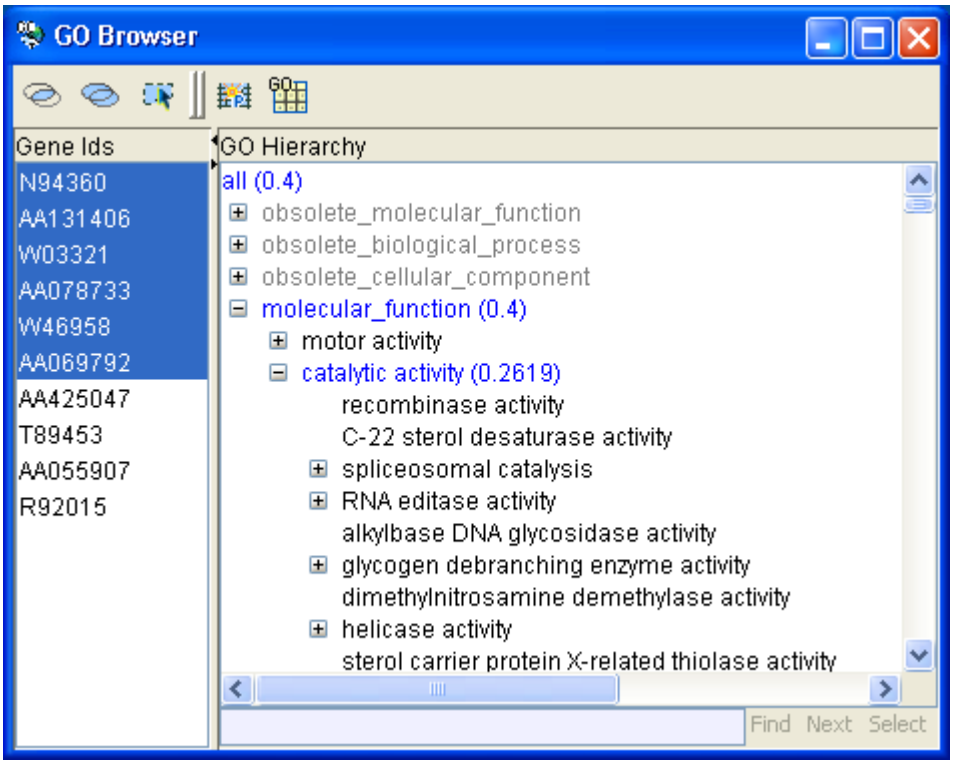


Figure 10.4: GO Browser Showing Gene Ontology terms for selected genes.

GO Browser

The GO Browser gives a visual representation of the Gene Ontology terms. A GO term is represented as a hierarchical structure in the **ArrayAssist** GO browser. On the left panel are the Gene Ids corresponding to the selected genes (the labels which appear here can be customized using Right-Click properties). The GO hierarchy appears on the panel on the right. The following operations are supported here.

- The functions on the GO Browser are explained below:
  - Double clicking on a GO term on the right panel will lasso all genes which have that term, in all lassoable views. Alternatively, click on a GO term and then click on Show Genes with This Term  icon to achieve the same effect.
  - Selecting genes from any view and then clicking on Show GO terms with significance  icon will highlight each term which is associated with at least one of the selected genes. In addition, the enrichment value of each GO terms that is represented in the selection will also be shown as a p-value. This can also be shown as two ratios, the first ratio shows the number of genes in the selection that have a particular GO terms to the total number of genes in the selection; and the second ratio shows the total number of genes in the dataset that have the GO term to the total number of genes in the dataset. You can change the way the enrichment value is represented in the GO Browser to a p-value or a ratio by Right-Click properties menu on the view.
  - Selecting genes from any view and then clicking on Show Common Terms  icon will highlight each term which is associated with all of the selected genes. In the *Matched Paths* tab, only the highlighted terms will appear, though not necessarily in the same order.

- Create a p-value Dataset:

  You can create a p-value dataset by Left-Click on the Create p-value dataset  icon. this will create a table with the GO terms, the number of genes in the selection with the GO term; the total number of genes in the selection; the number of genes with the GO term in the whole dataset; the total number of genes in the dataset; and the p-value for each GO term in the dataset. This table can then be exported and separately analysed.

- Create selected genes Vs. GO terms dataset

  You can create a dataset with selected genes based on the enrichment value or p-value cut-off. To create a dataset of selected genes that satisfy a p-value criteria, click on the Create selected genes Vs. GO terms dataset  icon. This will pop-up a dialog to enter the cut-off p-value. Enter a value between 0 and 1.0 and click OK. This will create a dataset with the selected genes that satisfy the p-value cut-off.

GO Computation

Suppose we have selected a subset of significant genes from a larger set and we want to classify these genes according to their ontological category. The aim is to see which ontological categories are important with respect to the significant genes. Are these the categories with the maximum number of significant genes, or are these the categories with maximum enrichment? Formally stated, consider a particular GO term G. Suppose we start with an array of n genes, m of which have this GO term G. We then identify x of the n genes as being significant, via a T-Test, for instance. Suppose y of these x genes have GO term G. The question now is whether there is enrichment for G, i.e., is y/x significantly larger than m/n. How do we measure this significance?

ArrayAssist computes a p-value to quantify the above significance. This p-value is the probability that a random subset of x genes drawn from the total set of n genes will have y or more genes containing the GO term G. This probability is described by a standard hypergeometric distribution (given n balls, m white, n-m black, choose x balls at random, what is the probability of getting y or more white balls). **ArrayAssist** uses the hypergeometric formula from first principles to compute this probability.

Finally, one interprets the p-value as follows. A small p-value means that a random subset is unlikely to match the actually observed incidence rate y/x of GO term G, amongst the x significant genes. Consequently, a low p-value implies that G is enriched (relative to a random subset of x genes) in the set of x significant genes.

NOTE: The same gene may be counted repeatedly in GO p-value computation due to association with multiple probesets. Currently, the computations don't take this factor into account.

Website	URL
Stanford SOURCE	http://genome-www5.stanford.edu/cgi-bin/SMD/source/sourceBatchSearch
UniGene	http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=unigene
LocusLink	http://www.ncbi.nlm.nih.gov/LocusLink/
NCBI-Nucleotide	http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Nucleotide
NCBI-BLAST (blastn)	http://www.ncbi.nlm.nih.gov/blast/Blast.cgi?PAGE=Nucleotides\&PROGRAM=blastn
NCBI-PubMed	http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=PubMed
SGD	http://db.yeastgenome.org/cgi-bin/SGD/

Table 10.2: Web Sites Used for Annotation

Chapter 11

The Genome Browser

ArrayAssist has an embedded genome browser which allows viewing of expression data juxtaposed against genomic features.

11.1 Genome Browser Usage

The genome browser is currently available from the Genome Browser link in the workflow browser. Clicking on this link will launch an empty genome browser and the *Tracks Manager* to choose the tracks to be displayed in the Genome Browser.

There are three kinds of tracks supported:

Static Tracks, *Data Tracks* and *Profile Tracks*. Static Tracks contain static information (i.e., unrelated to data) on genomic features, typically genes, exons and introns. Data Tracks display data from any chosen dataset in the project open currently; these tracks are meant to visualize genes, with each gene represented by a rectangle drawn from the chromosomal start location to the chromosomal stop location, and overlapping rectangles staggered out. Profile Tracks display data from any chosen dataset in the project open currently as well; these tracks are meant to visualize signal profiles with each data point represented by a single dot at the chromosomal start location. Data Tracks present genes handling overlaps and handling strand information; profile tracks on the other hand are more suitable for viewing SNP information, e.g., copy numbers, LOH scores etc.

Information for Static Tracks. Statics track packages are available for Humans, Mice and Rats. For each of these organisms, there are multiple static track packages available: one called KnownGenes derived from the Table Browser at UCSC (which in turn is derived from RefSeq and GenBank,

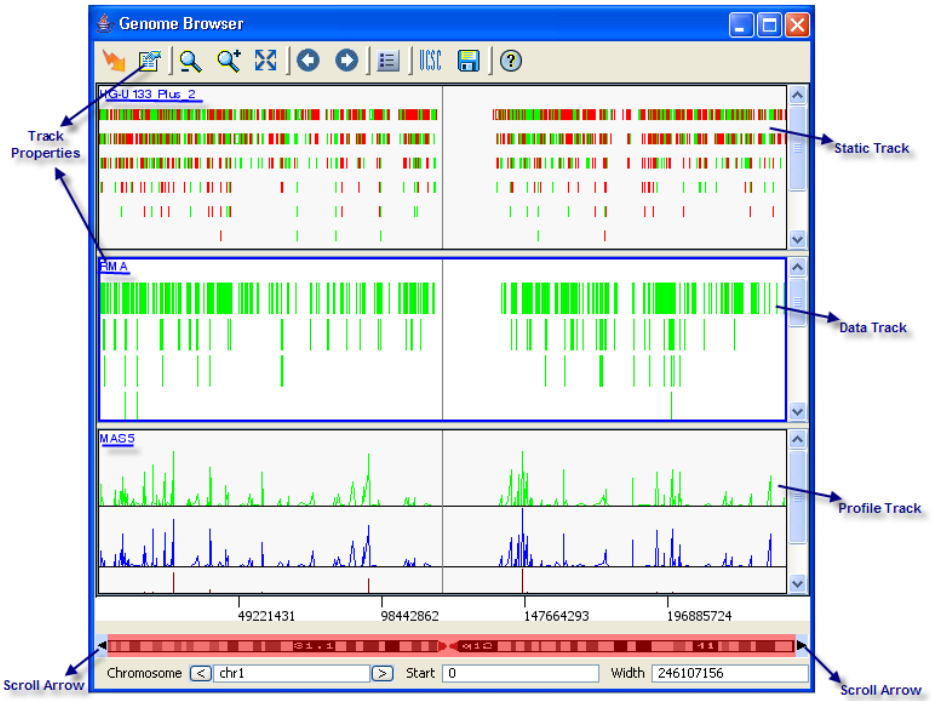


Figure 11.1: Genome Browser

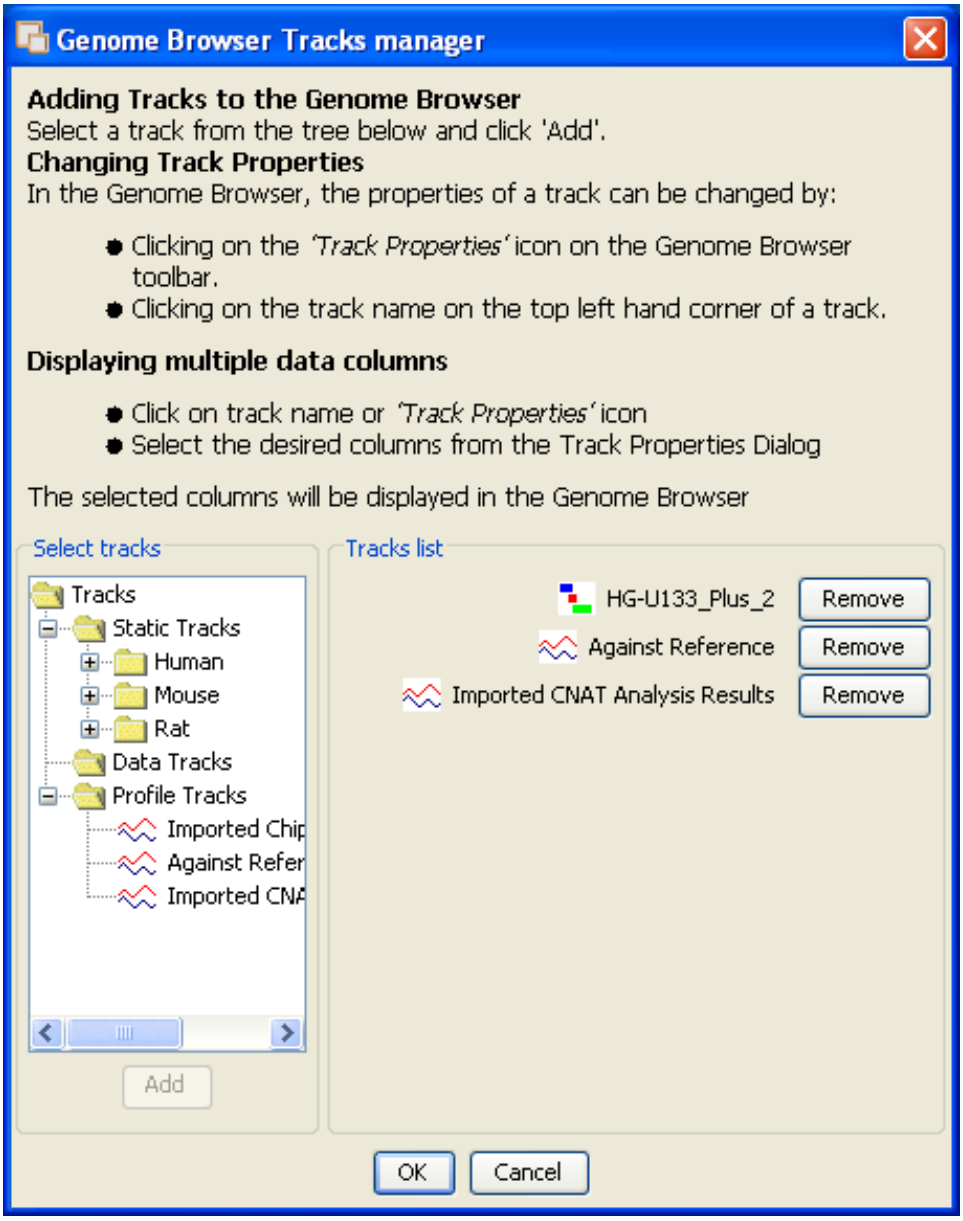


Figure 11.2: Tracks Manager

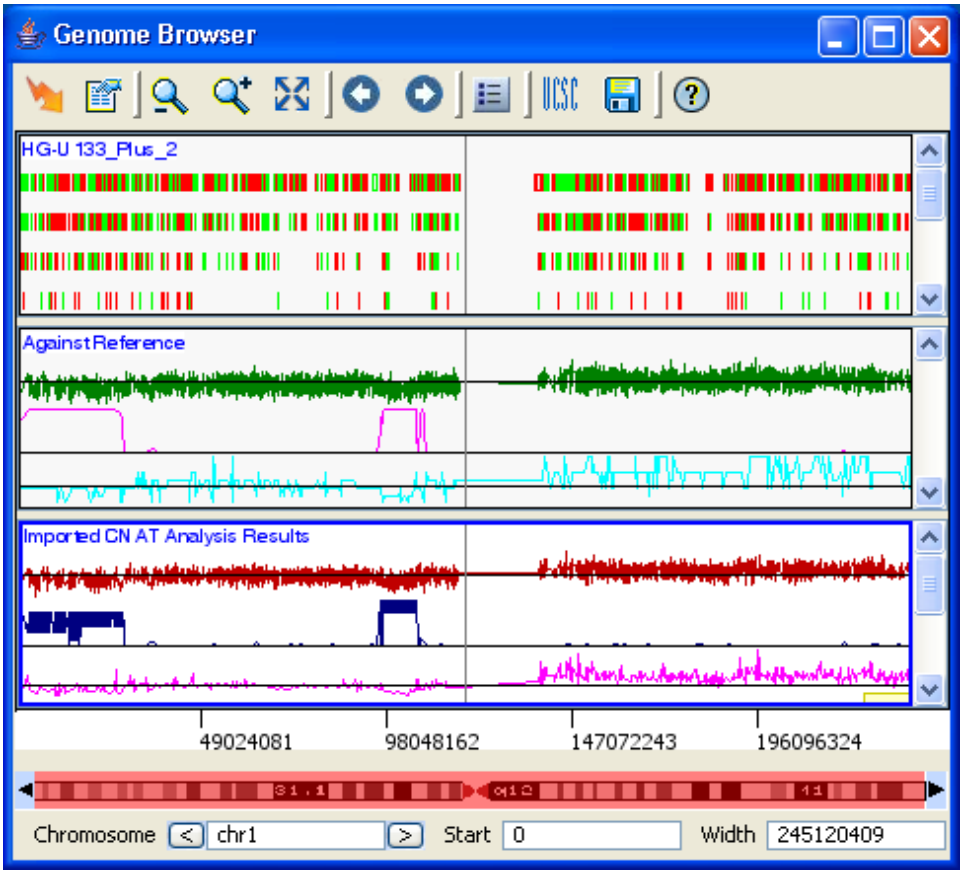


Figure 11.3: Profile Tracks in the Genome Browser

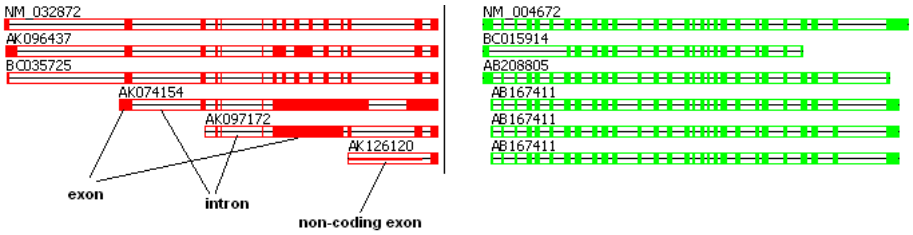


Figure 11.4: The KnownGenes Track

the latest versions available from the table browser at the time of the release are available, these are dated May 2004 for Humans, June 2003 for Rat, and Aug 2005 for Mouse) and another called Affymetrix ExonChip Transcripts derived from NetAffx annotations for the Exon chips. In addition, for Humans, there is an HG_U133Plus_2 static track as well. Each package can be downloaded using *Tools* $\longrightarrow$*Data Updates*, look for the genome browser package for the organism of interest. Specific static track packages for other organisms are available on demand.

Adding/Removing Tracks. Click on the TracksManager  icon.This will show a view in which all available tracks will be listed in the panel on the left. Static tracks for which the genome browser package has been downloaded as described above will appear in the list of static tracks. As regards Data Tracks, all open datasets in the project which appear in the navigator and which contain chromosome number, start, stop and strand columns will appear in the list of data tracks. Select a track of your interest and click on the Add button. After a brief delay, this track will be shown on the right. Removing this track at a later point is easily done by clicking on the Remove button. Multiple tracks can be added to the browser, though one at a time. The recommended number of tracks in the browser at any given time is at most 3 for efficiency.

Requirements for a Data Track. Note that to create a data track corresponding to a particular dataset in your project, you need to have 4 special columns with the following marks: chromosome number, chromosome start index, chromosome end index, and strand. If you do not have these columns but these are present in some other dataset you can use either the *Import Annotations* function in the workflow browser or the *Data*$\longrightarrow$*Columns*$\longrightarrow$*ImportColumns* function to import these columns from an external file. After you do this remember to mark these columns us-

ing *Data⟶Data Properties* with the appropriate marks. Note that for Affymetrix projects, all these columns will be there and marked by default (except for older projects created prior to April 06 for which users will need to download the new library packages and then do the *Import Annotation* step.

Requirements for a Profile Track. Note that to create a profile track corresponding to a particular dataset in your project, you need to have 2 special columns with the following marks: chromosome number and chromosome start index. If you do not have these columns but these are present in some other dataset you can use either the *Import Annotations* function in the workflow browser or the *Data⟶Columns⟶ImportColumns* function to import these columns from an external file. After you do this remember to mark these columns using *Data⟶Data Properties* with the appropriate marks. Note that for all Affymetrix projects, all these columns will be there and marked by default (except for older projects created prior to April 06 for which users will need to download the new library packages and then do the *Import Annotation* step.

Track Layout. Data tracks are separated by chromosome strand with the positive strand appearing at the top and negative strand at the bottom. Static and Profile tracks are not separated by chromosome strand. In static tracks, transcripts are colored red for the positive strand and green for the negative strand.

Track Properties. To set track properties, click on the track name, which is present at the top left of the corresponding track. Alternatively, first select the track. The selected track will be indicated by a dark blue outline. Click on the Track Properties  iconon the tool bar of the Genome Browser. This opens a dialog which allows setting labels on Static tracks, colors, labels and heights on Data Tracks, and enables importing data columns and setting colors on Profile Tracks. Data tracks can be colored/labelled/heighted by any relevant column in the corresponding dataset. Colors in the profile track can be changed by going to *Change Track Properties ⟶Rendering*. Profile Static tracks can be colored/labelled by only the supplied set of features and not by data.

Note that the Height By property on data tracks works as follows. If the selected column to height by has only positive values then all heights will be scaled so the maximum value has the max-height specified; all features will be drawn facing upwards on a fixed base line. If all values are negative, then heights are scaled as above but features are drawn downwards from a fixed baseline. If the selected column has both negative and positive values,

then the scaling is done so that the maximum absolute value in the column is scaled to half the max-height specified and features are drawn upwards or downwards appropriately on a central baseline. Also note that increasing the max-height parameter beyond a point does cause one or both tracks to go out of view at this point and will be fixed in a future release.

Profile Tracks allow viewing of multiple selected columns in the same track; each column is displayed as a profile whose height is adjustable based on the height parameter in the properties dialog. Profiles for all selected columns can be viewed on top of each other or staggered out, by checking the check-box in the properties dialog. In addition, profiles can also be smoothed by providing the length of the smoothing window (a value of x will average over a window of size $x/2$ on either side).

Both Data and Static track features show details on mouseover; the details shown are exactly those provided by the Label By property. Note that if a feature is not very wide then a label for it is not shown but the mouseover will work nevertheless. Profile tracks show the actual profile value on mouseover.

Zooming into Regions of Interest. First, by entering appropriate numbers in the text boxes at the bottom, you can select a particular chromosome, and a window in that chromosome. Another way to zoom in is to right click and go to Zoom Mode and then draw a rectangle with the mouse to zoom into a specified region. Yet another way is to use the zoom in and out icons on the genome browser toolbar. Further, the red bar and the bottom can be dragged to scroll across the length of the chromosome. Sometimes if it has become too thin, then you will need to zoom out till it becomes thick enough to grab with a mouse and drag. Finally, the arrows at the left and right bottom can also be used to scroll.

Selections. You can select features in any data track by going to selection mode on the right-click menu and dragging a region around the features of interest. All corresponding rows will be selected in the corresponding dataset and also lassoed to all open datasets and views. Conversely, if you have rows selected in any dataset and you wish to focus on the corresponding features in a particular data track of the browser, then click on the NextSelected  icon or the PrevSelected  icon; the next/previous feature selected in the data track will be brought to focus on the vertical centerline. Note that sometime this feature may not be visible because of fractional width, in which case zooming in will show the feature. Additionally, note that if there are multiple data tracks then the above icons will move to the next/previous item selected in the topmost of these data tracks.

Exporting Figures. All profiles within the active track (as indicated by the blue outline) can be exported using the Export As Image feature in the right-click menu. The image can be exported in a variety of formats, the .jpg, .jpeg, .png,.bmp and .tiff. By default, the image is exported as an anti-alias (high-quality). For details regarding the print size and image resolution, see the chapter on visualization

Creating Gene Lists. Use Save Selection in Active Track as GeneList  icon to create a gene list with the items visible on the currently active track (click on the track to make it active). A new gene list will appear in the gene list interface.

Saving BED files. Use Save Selection as Text  icon to create a BED file containing selected chromosomal locations in the active track.

Linking to the UCSC Browser. Clicking on the UCSC icon icon on the toolbar will open the UCSC genome browser in a web browser window at the current location. Note that the default organism for this link is assumed to be human. If you have a different organism of interest, edit the UCSC URL appropriately in *Tools* $\longrightarrow$ *Options*.

Chapter 12

Clustering: Identifying Rows with Similar Behavior

12.1 What is Clustering

Cluster analysis is a powerful way to organize rows in the dataset into groups or clusters of similar rows. There are several ways of defining the similarity measure, or the distance between two rows. While some methods are purely mathematical, others use domain specific knowledge about the rows. The Euclidean measure is the most commonly used measure, though several other measures are in use as well.

ArrayAssist's clustering module offers the following unique features:

- A variety of clustering algorithms: K-Means, Hierarchical, EigenValue, Self Organizing Maps (SOM), Random Walk, and Principal Components Analysis (PCA) clustering, along with a variety of distance functions - Euclidean, Square Euclidean, Manhattan, Chebychev, Differential, Pearson Absolute and Pearson Centered. Data is sorted on the basis of such distance measures to group both rows and columns into most similar clusters. Since different algorithms work well on different kinds of data, this large battery of algorithms and distance measures ensures that a wide variety of data can be clustered effectively.

- A variety of interactive views such as the ClusterSet View, the Dendrogram View and the Similarity Image View are provided for visualization of clustering results. These views allow drilling down into subsets of data and collecting together individual rows or groups of rows which look interesting into new datasets for further analysis. All

views as lassoed, and enable visualization of a cluster in multiple forms based on the number of different views opened.

12.2 Clustering Pipeline

The typical sequence of operations to be followed before and during cluster analysis is as follows:

1. Load data into **ArrayAssist**.

   The loading of data is described in Loading Data.

2. Preprocess the data to remove missing values. All input to clustering algorithms needs to be free of (so either remove or filter missing values). Some distance measures depend on the range of data in each dimension and, therefore, input data can be optionally normalized to lie in the same range.

   The procedure for removing rows with missing values is described in Dataset Operations.

3. Cluster the data using the appropriate algorithm and distance measure. Data can be clustered along rows and along columns simultaneously (except when using the SOM clustering method); (NOTE: the same algorithm and parameters will be used in both clusterings).

   To cluster the data, click *Cluster* in the menu bar and choose a suitable clustering algorithm from the drop down menu.

4. View clustering results. Some algorithms directly generate clusters as their result (these include K-Means, EigenValue, SOM and PCA clustering) while others (e.g. Hierarchical, SOM and Random Walk) generate relationship trees which are shown as dendrograms and on which cutoffs need to be applied to obtain discrete clusters.

5. Once clusters are identified, cluster names can either be appended to the dataset, or new subsets of clustered data can be created for further analysis. These subsets can be created either by copying selected rows to the *Clipboard* or by using the *Create New Dataset* feature on the selected rows in each of the interactive views.

Note: Clustering works on all continuous numeric columns by default in the absence of any column selection. The identifier and class-label column

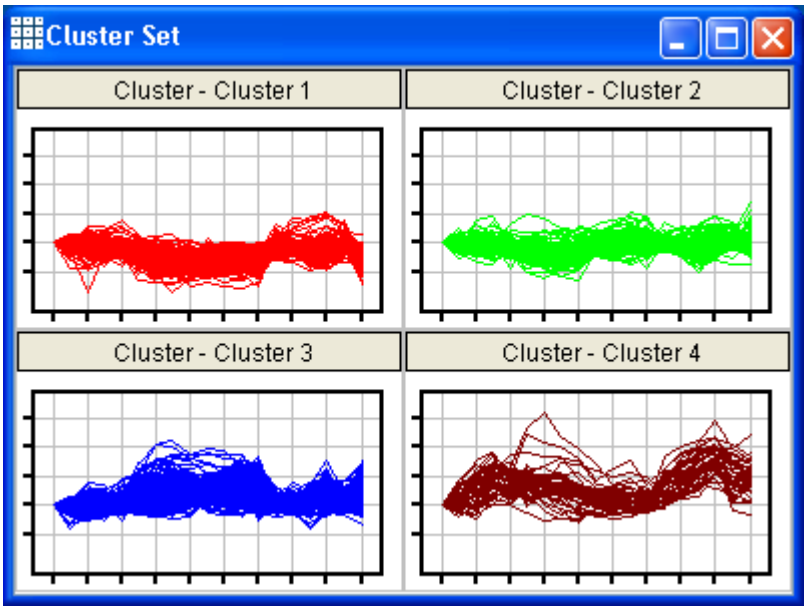


Figure 12.1: Cluster Set from K-Means Clustering Algorithm

are omitted by default. To run clustering on only a desired exact subset of the columns, choose appropriate columns from the *Columns* tab in the *Clustering Parameters* input dialog.

12.3 Graphical Views of Clustering Analysis Output

ArrayAssist incorporates a number of rich and intuitive graphical views of clustering results. All the views are highly interactive. Clusters and other data of interest can be picked out with ease to create new datasets, or rows of interest can be copied to the clipboard.

12.3.1 Cluster Set

Algorithms like K-Means clustering generate a fixed number of clusters. The Cluster Set plot graphically displays high-level overview information of all clusters in the data. Every cluster is represented by the average of expression profile of all rows in that cluster (light green line by default), along with the

minimum and maximum deviation around the mean in each column (black vertical lines). Clusters are labeled as *Cluster 1*, *Cluster 2* ... and so on. The heading also indicates the number of rows contained in the cluster.

Some datasets tend to generate many small clusters containing only a few rows each, in addition to large clusters. Small clusters, which account for less than 5 percent of the total number of rows each, are not plotted separately. Instead, they are grouped together in a residual cluster plot, where all rows from such clusters are plotted in a single cluster set labeled as *n Small Clusters*.

Cluster Set Operations

The Cluster Set view is a lassoed view and can be used to extract meaningful data for further use. The current lasso is displayed as a background color change in every individual cluster. The *level* of the background painted in selection color indicates the fraction of the rows contributed to the current lasso from the individual clusters.

Lasso Left click on an individual cluster to select all rows in the cluster. These rows are highlighted in all other lassoable views open currently. This also acts as a useful way to crosscheck the cluster quality with other clustering outputs like the dendrogram and the similarity image. NOTE: The background of the selected cluster changes to selection color indicating that all rows in the cluster have been lassoed.

View Gene Profiles in a Cluster Double-click on an individual cluster to bring up a Profile plot of the rows in the cluster. The entire range of functionality of the Profile view is then available for extraction of useful data.

Export Cluster Names to Dataset It is possible to export the clustering information back to the dataset by right clicking on the cluster set plot and choosing *Export Column to Data Set*. This operation appends a new column to the dataset, with the appropriate cluster name for each row in the dataset.

Cluster Set Properties

The properties of the Cluster Set Display can be altered by right clicking on the Cluster Set View and choosing *Properties* from the drop down menu.

The Cluster Set view, similar to the main menu bar, supports the following configurable properties:

Rendering The rendering of the fonts, colors and offsets on the Profile Plot can be customized and configured.

Fonts: All fonts on the plot, can be formatted and configured. To change the font in the view, Right-Click on the view and open the Properties dialog. Click on the Rendering tab of the Properties dialog. To change a Font, click on the appropriate drop-down box and choose the required font. To customize the font, click on the customize button. This will pop-up a dialog where you can set the font size and choose the font type as bold or italic.

Special Colors: All the colors that occur in the plot can be modified and configured. The plot Background Color, the Axis Color, the Grid Color, the Selection Color, as well as plot specific colors can be set. To change the default colors in the view, Right-Click on the view and open the Properties dialog. Click on the Rendering tab of the Properties dialog. To change a color, click on the appropriate color bar. This will pop-up a Color Chooser. Select the desired color and click OK. This will change the corresponding color in the View.

Offsets: The left offset, right offset and the top offset and bottom offset of the plot can be modified and configured. These offsets may be need to be changed if the axis labels or axis titles are not completely visible in the plot, or if only the graph portion of the plot is required. To change the offsets, Right-Click on the view and open the Properties dialog. Click on the Rendering tab. To change plot offsets, move the corresponding slider, or enter an appropriate value in the text box provided. This will change the particular offset in the plot.

Quality Image The Profile Plot image quality can be increased by checking the High-Quality anti-aliasing option. This is slow however and should be used only while printing or exporting the Profile Plot.

Columns The Profile Plot is launched with a default set of columns. The set of visible columns can be changed from the Columns tab. The columns for visualization and the order in which the columns are visualized can be chosen and configured for the column selector. Right-Click on the view and open the properties dialog. Click on the columns tab. This will open the column selector panel. The column selector

panel shows the Available items on the left-side list box and the Selected items on the right-hand list box. The items in the right-hand list box are the columns that are displayed in the view in the exact order in which they appear.

To move a columns from the Available list box to the Selected list box, highlight the required items in the Available items list box and click on the right arrow in between the list boxes. This will move the highlighted columns from the Available items list box to the bottom of the Selected items list box. To move columns from the Selected items to the Available items, highlight the required items on the Selected items list box and click on the left arrow. This will move the highlight columns from the Selected items list box to the Available items list box in the exact position or order in which the column appears in the dataset.

You can also change the column ordering on the view by highlighting items in the Selected items list box and clicking on the up or down arrows. If multiple items are highlighted, the first click will consolidate the highlighted items (bring all the highlighted items together) with the first item in the specified direction. Subsequent clicks on the up or down arrow will move the highlighted items as a block in the specified direction, one step at a time until it reaches its limit. If only one item or contiguous items are highlighted in the Selected items list box, then these will be moved in the specified direction, one step at a time until it reaches its limit. To reset the order of the columns in the order in which they appear in the dataset, click on the reset icon next to the Selected items list box. This will reset the columns in the view in the way the columns appear in the view.

To highlight items, Left-Click on the required item. To highlight multiple items in any of the list boxes, Left-Click and Shift-Left-Click will highlight all contiguous items, and Left-Click and Ctrl-Left-Click will add that item to the highlight elements.

The lower portion of the Columns panel provides a utility to highlight items in the Column Selector. You can either match by Name or by Experimental Factor (if specified). To match by Name, select Match By Name from the drop down list, enter a string in the Name text box and hit Enter. This will do a substring match with the Available List and the Selected list and highlight the matches. To match by Experiment Grouping, the Experiment Grouping information must be

provided in the dataset. If this is available, the Experiment Grouping drop down will show the factors. The groups in each factor will be show in the Groups list box. Selecting specific Groups from the text box will highlight the corresponding items in the Available items and Selected items box above. These can be moved as explained above. By default, the match By Name is used.

Description The title for the view and description or annotation for the view can be configured and modified from the description tab on the properties dialog. Right-Click on the view and open the Properties dialog. Click on the Description tab. This will show the Description dialog with the current Title and Description. The title entered here appears on the title bar of the particular view and the description if any will appear in the Legend window situated in the bottom of panel on the right. These can be changed changing the text in the corresponding text boxes and clicking OK. By default, if the view is derived from running an algorithm, the description will contain the algorithm and the parameters used.

Trellis The Profile Plot can be trellised based on a trellis column. To trellis the Profile Plot, click on Trellis on the Right-Click menu or click Trellis from the View menu. This will launch multiple Profile Plot in the same view based on the trellis column. By default the trellis will be launched with the categorical column with the least number of categories in the current dataset. You can change the trellis column by the properties of the trellis view.

Axes The grids, axes labels, and the axis ticks of the plots can be configured and modified. To modify these, Right-Click on the view, and open the Properties dialog. Click on the Axis tab. This will open the axis dialog. The plot can be drawn with or without the grid lines by clicking on the show grids option. The tics and axis labels are automatically computed for the plot and show on the plot. You can show or remove the axis labels by clicking on the Show Axis Labels check box. The number of ticks on the axis are automatically computed to a show equal intervals between the minimum and maximum and displayed. You can increase the number of ticks displayed on the plot by moving the Axis Ticks slider. For continuous data columns, you can double the number of ticks shown by moving the slider to the maximum. For categorical columns, if the number of categories are less than ten, all the categories are show and moving the slider does not increase the

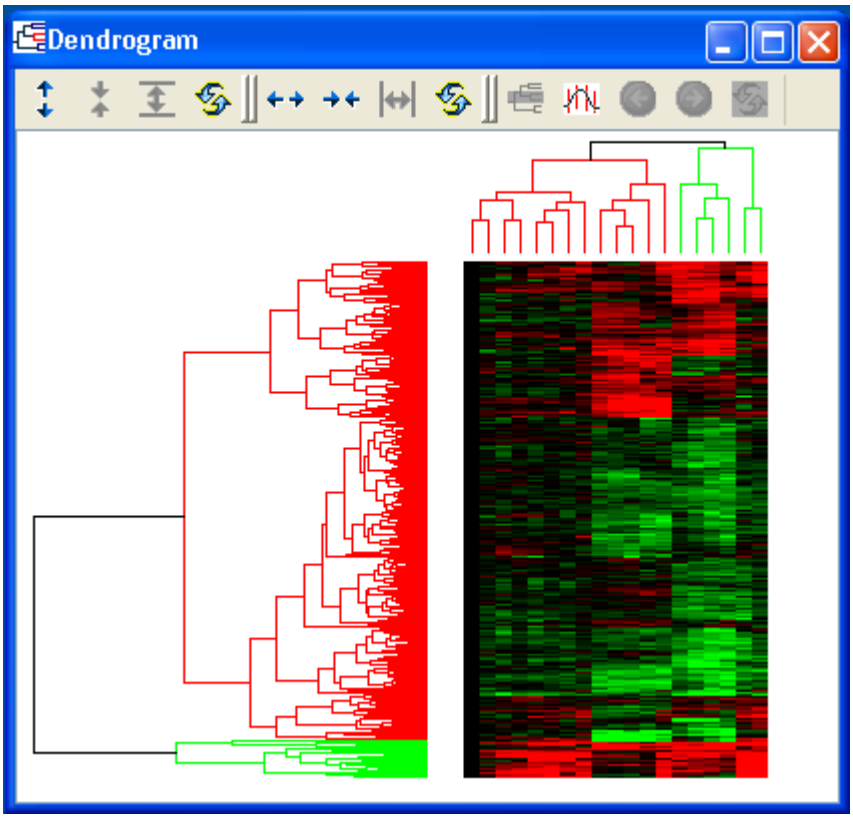


Figure 12.2: Dendrogram of Hierarchical Clustering

number of tics.

Visualization

Color Each point can be assigned either a fixed customizable color or a color based on its value in a specified column. The Customize button can be used to customize colors for both the fixed and the By-Column options.

In the cluster set plots, a mean profile can be drawn by selecting the box named *Display mean profile*.

12.3.2 Dendrogram

Some clustering algorithms like Hierarchical Clustering do not distribute data into a fixed number of clusters, but produce a grouping hierarchy. Most similar rows are merged together to form a cluster and this combined entity is treated as a unit thereafter. The result is a tree structure or a dendrogram, where the leaves represent individual rows and the internal nodes represent clusters of similar rows.

The leaves are the smallest clusters with one gene each. Each node in the tree defines a cluster. The distance at which two clusters merge (a measure of dissimilarity between clusters) is called the threshold distance, which is measured by the height of the node from the leaf. Every gene is labeled by its identifier as specified by the id column in the dataset. A Heat Map is also included in the plot, with the rows permuted in the same order as they are in the dendrogram. This helps in visual confirmation of the clustering results.

When both rows and columns are clustered, the plot includes two dendrograms - the vertical dendrogram for rows, and the horizontal one for columns. Each of these can be manipulated independently.

When a clustering algorithm is run that allows for a dendrogram view, a new window is displayed in the desktop. The title of the window gives the name of the clustering algorithm that generated this dendrogram view, for example, *Hierarchical - Dendrogram*. The center of the window has the Heat map. Row labels are on the left and column labels on top, each with its respective dendrogram.

Dendrogram Operations

The dendrogram is a lassoed view and can be navigated to get more detailed information about the clustering results. Dendrogram operations are also available by Right-Click on the canvas of the Dendrogram. Operations that are common to all views are detailed in the section Common Operations on Table Views above. In addition, some of the heat specific operations and the Dendrogram properties are explained below:

Cell information in the Heat Map Mouse over any cell to get its expression value as a tool tip.

Lasso individual rows Select rows by clicking and dragging on the heat map or the row labels. It is possible to select multiple rows and intervals using *Shift* and *Control* keys along with mouse drag. The lassoed

rows are indicated in a light blue overlay.

Column Selection When Hierarchical clustering is executed on columns, columns can also be selected just like rows. Only the selected columns and rows are highlighted (and not the entire row). Note that when a dataset is created from the selection, only those columns that are selected will be in the new dataset along with all string and categorical columns.

Lasso Subtree in Dendrogram To select a sub-tree from the dendrogram, left-click close to the root node for this sub-tree but within the region occupied by this sub-tree. In particular, left-clicking anywhere will select the smallest sub-tree enclosing this point. The root node of the selected sub-tree is highlighted with a blue diamond and the sub-tree is marked in bold. Note that when a dataset is created from the selection, only those columns that are selected will be in the new dataset along with all string and categorical columns.

Zoom Into Subtree Left-click in the currently selected sub-tree again to redraw the selected sub-tree as a separate dendrogram. The heat map is also updated to display only the rows (or columns) in the current selection. This allows for drilling down deeper into the tree to the region of interest to see more details.

Export As Image: This will pop-up a dialog to export the view as an image. This functionality allows the user to export very high quality image. You can specify any size of the image, as well as the resolution of the image by specifying the required dots per inch (dpi) for the image. Images can be exported in various formats. Currently supported formats include png, jpg, jpeg, bmp or tiff. Finally, images of very large size and resolution can be printed in the tiff format. Very large images will be broken down into tiles and recombined after all the images pieces are written out. This ensures that memory is but built up in writing large images. If the pieces cannot be recombined, the individual pieces are written out and reported to the user. However, tiff files of any size can be recombined and written out with compression. The default dots per inch is set to 300 dpi and the default size if individual pieces for large images is set to 4 MB. These default parameters can be changed in the tools $\longrightarrow$Options dialog under the **Export as Image**

The user can export only the visible region or the whole image. Images

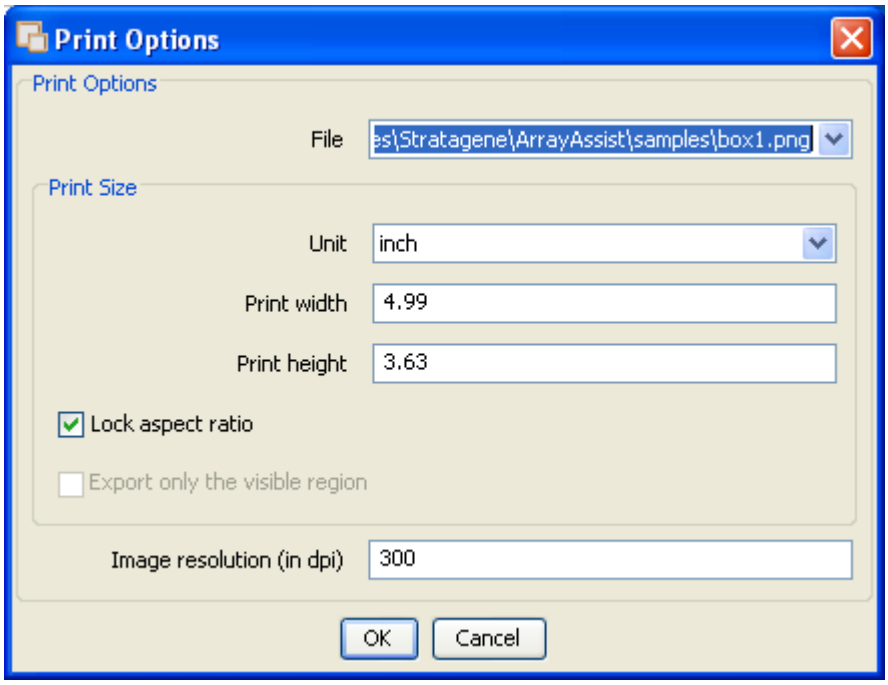


Figure 12.3: Export Image Dialog

of any size can be exported with high quality. If the whole image is chosen for export, however large, the image will be broken up into parts and exported. This ensures that the memory does not bloat up and that the whole high quality image will be exported. After the image is split and written out, the tool will attempt to combine all these images into a large image. In the case of png, jpg, jpeg and bmp often this will not be possible because of the size of the image and memory limitations. In such cases, the individual images will be written separately and reported. However, if a tiff image format is chosen, it will be exported as a single image however large. The final tiff image will be compressed and saved.

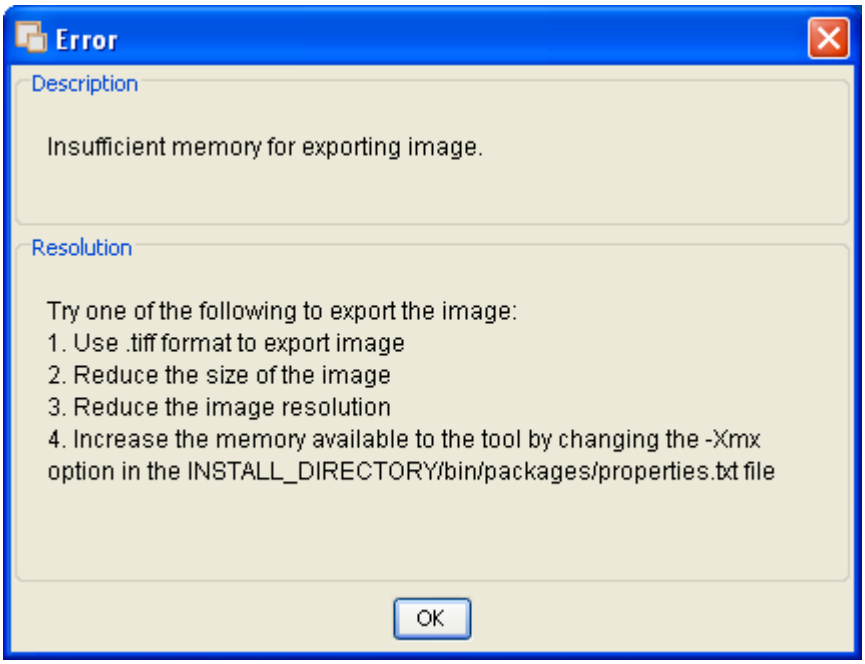

Error
Description
Insufficient memory for exporting image.
Resolution
Try one of the following to export the image:
1. Use .tiff format to export image
2. Reduce the size of the image
3. Reduce the image resolution
4. Increase the memory available to the tool by changing the -Xmx
option in the INSTALL_DIRECTORY/bin/packages/properties.txt file
OK

Figure 12.4: Error Dialog on Image Export

Note: This functionality allows the user to create images of any size and with any resolution. This produces high-quality images and can be used for publications and posters. If you want to print vary large images or images of very high-quality the size of the image will become very large and will require huge resources. If enough resources are not available, an error and resolution dialog will pop us, saying the image is too large to be printed and suggesting you to try the tiff option, reduce the sixe of image or resolution of image, or to increase the memory avaliable to the tool by changing the -Xmx option in INSTALL_DIR/bin/packages/properties.txt file.

Note: You can export the whole dendrogram as a single image with any size and desired resolution. To export the whole image, choose this option in the dialog. The whole image of any size can be exported as a compressed tiff file. This image can be opened on any machine with enough resources for handling large image files.



Figure 12.5: Dendrogram Toolbar

Export as HTML: This will export the view as a html file. Specify the file name and the the view will ve exported as a HTML file that can be viewed in a browser and deployed on the web. If the whole image export is chosen, multiple images will be exported and can be opened in composed and open in a browser.

Dendrogram Toolbar

The dendrogram toolbar offers the following functionality:



Mark Clusters: This functionality allows marking the current selected subtree with a user-specified label, as well as coloring the subtree with a color of choice to graphically depict different subtrees corresponding to different clusters in separate colors. This information can subsequently used to create a *Cluster Set* view where each marked subtree appears as an independent cluster.



Create Cluster Set: This operation allows the creation of clusters from the dendrogram in two ways:

- Using marking information generated by the step described above, and creating a separate cluster for each marked subtree. Select the *Use Marked Nodes* checkbox and click on OK. This will produce as many clusters as there are marked subtrees. All unmarked rows will but put in a residual cluster called *'remaining'*.
- by giving a choice of a threshold distance at which rows are considered to form a cluster. Move the slider to move the threshold-distance line in the dendrogram. All subtrees where the threshold distance is less than the distance specified by the red line will be marked with a red diamond, indicated that a cluster has been induced at that distance. Click on OK to generate a Cluster Set view of the data.



Navigate Back: Click to navigate to previously selected subtree.



Navigate Forward: Click to navigate to current (or next) selected subtree.



Reset Tree Navigation: Click to reset the display to the entire tree.

Zoom in rows: Click to increase the dimensions of the dendrogram. This increases the separation between two rows at the leaf level. Row labels appear once the separation is large enough to accommodate label strings.



Zoom out rows: Click to reduce dimensions of the dendrogram so that leaves are compacted and more of the tree structure is visible on the screen. The heat map is also resized appropriately.



Fit rows to screen: Click to scale the dendrogram to fit entirely in the window. This is useful in obtaining an overview of clustering results for a large dendrogram. A large image, which needs to be scrolled to view completely, fails to effectively convey the entire picture. Fitting it to the screen gives a quick overview.



Reset row zoom: Click to scale the dendrogram back to default resolution. It also resets the root to the original entire tree.
Note: Row labels are not visible when the spacing between leaf nodes becomes too small to display labels. Zooming in or Resetting will restore these.



Zoom in columns: Click to scale up the column dendrogram.



Zoom out columns: Click to reduce the scale of the column dendrogram so that leaves are compacted and more of the tree structure is visible on the screen. The heat map is also resized appropriately.



Fit columns to screen: Click to scale the column dendrogram to fit entirely in the window. This is useful in obtaining an overview of clustering results for a large dendrogram. A large image, which needs to be scrolled to view completely, fails to effectively convey the entire picture. Fitting it to the screen gives a quick overview.



Reset columns zoom: Click to scale the dendrogram back to default resolution. It also resets the root to the original entire tree.

Note: Column Headers are not visible when the spacing between leaf nodes becomes too small to display labels. Zooming or Resetting will restore these.

Dendrogram Properties

The Dendrogram view supports the following configurable properties:

Color and Saturation Threshold Settings To access these settings, click on the dendogram and select *Properties* from the drop down menu, and click on *Visualization*. Allows changing the minimum, maximum and middle colors as well the threshold values for saturation. Saturation control enables detection of subtle differences in gene expression levels for those rows, which do not exhibit extreme levels of under or over-expression. Move the sliders to set the saturation thresholds; alternatively, the values can be provided in the textbox next to the slider. Please note that if you type values into the text box, you will have to hit Enter for the values to be accepted.

Label Rows by Allows the choice of a column whose values are used to label the rows in the dendrogram. Identifier column is used to label rows by default if defined.

Size Settings Allows changing the size of the row and column headers, as well the row and column dendrograms. To change the size settings, Move the sliders to see the underlying view change.

Description Clicking on the *Description* under *Properties* displays the title and parameters of the clustering algorithm used.

12.3.3 Similarity Image

The Similarity Image is an image-based, intuitive view of the clustering results and gives a good indication of the quality of clustering. Every clustering algorithm permutes the rows to bring together similar rows and place the dissimilar ones apart. The similarity between these permuted sequences of rows is plotted as a 2D gray-scale image. It is laid out as a symmetric

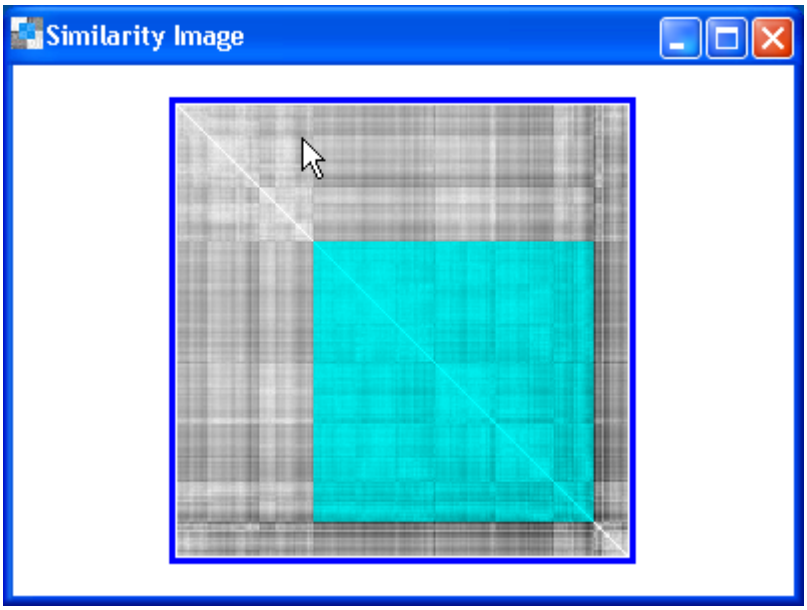


Figure 12.6: Similarity Image from Eigen Value Clustering Algorithm

grid with rows along the rows and the columns; the brightness of pixel i, j is a measure of similarity between gene i and gene j. Diagonals are the brightest, indicating maximum similarity, that of a gene with itself. For good clustering results, the image will show tight white squares along the diagonal, while being dark in other regions. This indicates that rows within clusters are highly similar, whereas rows across clusters are very dissimilar. Sometimes clustering algorithms will split a cluster into one or more pieces. This can be spotted easily on the image. The off-diagonal blocks for these pieces will also be white indicating a split of clusters.

Note: For very large datasets, the Similarity Image view would produce huge images with large memory overheads. To reduce this demand, the image is down-sampled and a maximum of 1024X1024 pixels are used.

Similarity Image Operations

The Similarity Image is a lassoed view and appears as a new window on the **ArrayAssist** desktop. All lassoed rows appear in a different background overlay color, and it is easy to identify whether they are part of a tight, compact cluster by checking that the lasso area lies completely in a single

cluster. The view can be manipulated in the following ways:

Cluster Selection Left-click at one end of the diagonal of the region to be selected. Drag along the diagonal to select the required region. A square with a boundary marking the selected region will be overlaid on the Similarity Image. The selected region is highlighted with a blue background, and all rows corresponding to the region are lassoed. Note that if more than 1024 elements are clustered the Similarity View will be a sampled image and will not be lassoable. Only Zoom Mode will be available for such an image.

Zoom Mode The view supports zooming in and out like other zoomable views in **ArrayAssist**. Switch to zoom mode by clicking the *Zoom/Selection Mode* toggle button in the toolbar (or using the right-click context menu). Select a region of interest by dragging a square outline while pressing the left mouse button. The view zooms to the region on interest and displays the selected region in the available window area.

Similarity Image Properties

The Similarity Image view supports the following configurable properties, which can be chosen by clicking *Visualization* under the *properties* menu.

Minimum Similarity Color Allows a choice of the color used to represent zero similarity. Default value is black.

Maximum Similarity Color Allows a choice of the color used to represent 100% similarity. Default value is white.

In addition to these configurable properties, clicking on the *Description* under *Properties* lists the type of algorithm and the parameters used.

12.3.4 U Matrix

The U-Matrix view is primarily used to display results of the SOM clustering algorithm. It is similar to the Cluster Set view, except that it displays clusters arranged in a 2D grid such that similar clusters are physically closer in the grid. The grid can be either hexagonal or rectangular as specified by the user. Cells in the grid are of two types, nodes and non-nodes. Nodes and non-nodes alternate in this grid. Holding the mouse over a node will cause that node to appear with a red outline. Clusters are associated only with nodes and each node displays the reference vector or the average expression

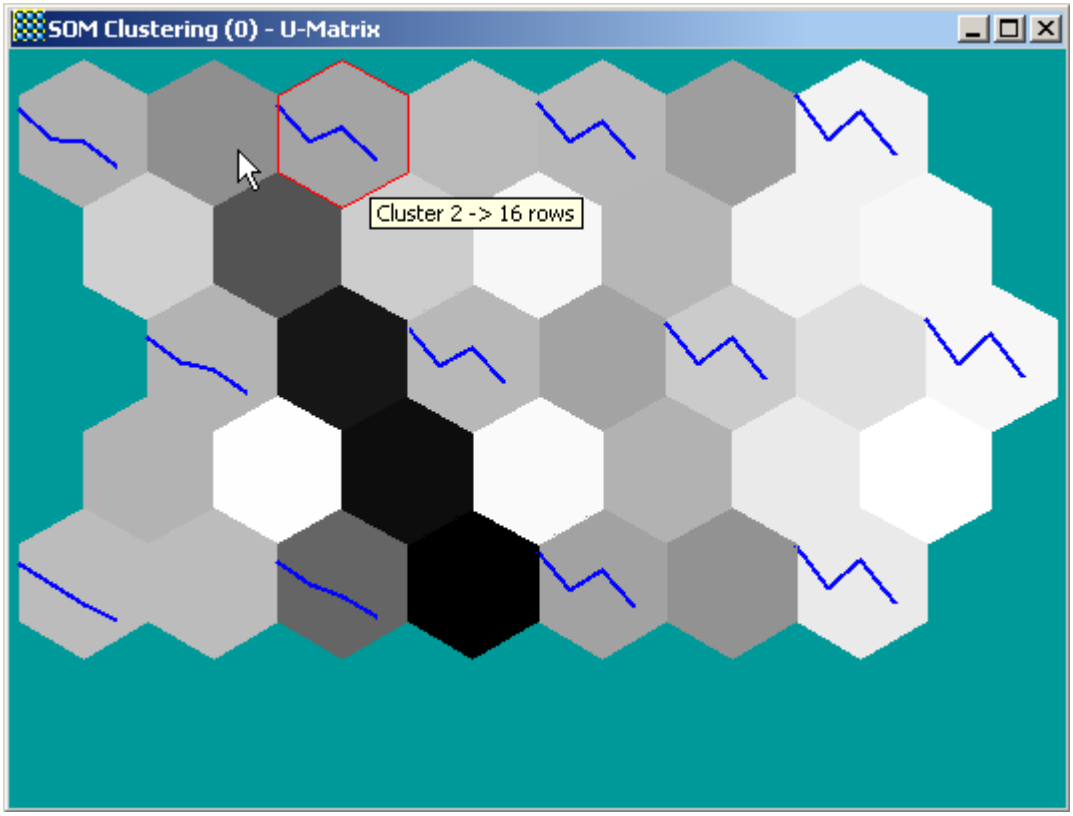


Figure 12.7: U Matrix for SOM Clustering Algorithm

profile of all rows mapped to the node. This average profile is plotted in blue. The purpose of non-nodes is to indicate the similarity between neighboring nodes on a grayscale. In other words, if a non-node between two nodes is very bright then it indicates that the two nodes are very similar and conversely, if the non-node is dark then the two nodes are very different. Further, the shade of a node reflects its similarity to its neighboring nodes. Thus not only does this view show average cluster profiles, it also shows how the various clusters are related. Left-clicking on a node will pull up the Profile plot for the associated cluster of rows.

U-Matrix Operations

The U-Matrix view supports the following operations.

Mouse Over Moving the mouse over a node representing a cluster (shown by the presence of the average expression profile) displays more information about the cluster in the tooltip as well as the status area. Similarly, moving the mouse over non-nodes displays the similarity between the two neighboring clusters expressed as a percentage value.

View Gene Profiles in a Cluster Left click on an individual cluster node to bring up a *Profile* view of the rows on the cluster. The entire range of functionality of the Profile view is then available.

U-Matrix Properties

The U-Matrix view supports the following properties which can be chosen by clicking *Visualization* under the *properties* menu.

High quality image An option to choose high quality image. Click on *Visualization* under *Properties* to access this.

Description Click on *Description* to get the details of the parameters used in the algorithm.

12.4 Distance Measures

Every clustering algorithm needs to measure the similarity (difference) between rows. Once a gene is represented as a vector in n-dimensional expression space, several distance measures are available to compute similarity. **ArrayAssist** supports the following distance measures:

- Euclidean: Standard sum of squared distance (L2-norm) between two rows.

$$\sqrt{\sum_i (x_i - y_i)^2}$$

- Squared Euclidean: Square of the Euclidean distance measure. This accentuates the distance between rows. Rows that are close are brought closer, and those that are dissimilar move further apart.

$$\sum_i (x_i - y_i)^2$$

- Manhattan: This is also known as the L1-norm. The sum of the absolute value of the differences in each dimension is used to measure the distance between rows.

$$\sum_i |x_i - y_i|$$

- Chebychev: This measure, also known as the L-Infinity-norm, uses the absolute value of the maximum difference in any dimension.

$$\max_i |x_i - y_i|$$

- Differential: The distance between two rows in estimated by calculating the difference in slopes between the expression profiles of two rows and computing the Euclidean norm of the resulting vector. This is a useful measure in time series analysis, where changes in the expression values over time are of interest, rather than absolute values at different times.

$$\sqrt{\sum_i [(x_{i+1} - x_i) - (y_{i+1} - y_i)]^2}$$

- Pearson Absolute: This measure is the absolute value of the Pearson Correlation Coefficient between two rows. Highly related rows give values of this measure close to 1, while unrelated rows give values close to 0.

$$\left| \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{(\sum_i (x_i - \bar{x})^2)(\sum_i (y_i - \bar{y})^2)}} \right|$$

- Pearson Centered: This measure is the 1-centered variation of the Pearson Correlation Coefficient. Positively correlated rows give values of this measure close to 1; negatively correlated ones give values close to 0, and unrelated rows close to 0.5.

$$\frac{\left[\left(\frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{(\sum_i (x_i - \bar{x})^2)(\sum_i (y_i - \bar{y})^2)}}\right) + 1\right]}{2}$$

The choice of distance measure and output view is common to all clustering algorithms as well as others like *Profile Matching* algorithms in **ArrayAssist**. In addition, for the EigenValue method alone, an additional distance measure (angular distance) is available.

- Angular This measure is similar to the Pearson Correlation coefficient except that the rows are not mean-centered. In effect, this measure treats the two rows as vectors and gives the cosine of the angle between the two vectors. Highly correlated rows give values close to 1, negatively correlated rows give values close to -1, while unrelated rows give values close to 0.

$$\frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2 \sum_i y_i^2}}$$

Finding Negatively Correlated Rows: All the above clustering methods and distance functions can be used to cluster together negatively correlated rows provided the data in the spreadsheet is ratio data in a logarithmic or related scale (e.g., the arcsinh scale). Use the Absolute feature on the spreadsheet to take the absolute values of the gene expressions and then use any of the above distance functions and clustering methods. The effect of this absolute feature can be undone post clustering if needed.

12.5 K-Means

This is one of the fastest and most efficient clustering techniques available, if there is some advance knowledge about the number of clusters in the data. Rows are partitioned into a fixed number (k) of clusters such that, rows within a cluster are similar, while those across clusters are dissimilar. To begin with, rows are randomly assigned to k distinct clusters and the average expression vector is computed for each cluster. For every gene, the algorithm then computes the distance to all expression vectors, and moves the gene to that cluster whose expression vector is closest to it. The entire process is repeated iteratively until no rows jump across clusters, or a maximum number of iterations is reached. K-Means clustering can be invoked by clicking on the Clustering menu and selecting K-Means. Clustering will be carried out on the current dataset in the Spreadsheet. The *Parameters* dialog box will appear. Various clustering parameters to be set are as follows:

Cluster On Dropdown menu gives a choice of Rows, or Columns, or Both rows and columns, on which clusters can be formed. Default is Rows.

Distance Metric Dropdown menu gives seven choices; Euclidean, Squared Euclidean, Manhattan, Chebychev, Differential, Pearson Absolute and Pearson Centered. The default is Euclidean.

Number of Clusters This is the value of k, and should be a positive integer. The default is 3.

Maximum Iterations This is the upper bound on the maximum number of iterations for the algorithm. The default is 50 iterations.

Views The graphical views available with K-Means clustering are -

- Cluster Set View
- Dendrogram View
- Similarity Image View.

Results of clustering will appear in the desktop, with each view as a separate window. K-Means and its output views will be added to the navigator.

Advantages and Disadvantages of K-Means: K-means is by far the fastest clustering algorithm and consumes the least memory. Its memory efficiency comes from the fact that it does not need a distance matrix. However, it tends to cluster in circles, so clusters of oblong shapes may not be identified correctly. Further, it does not give relationship information for rows within a cluster. When clustering with large datasets (say more than 7000 to 8000 rows on a 256MB RAM machine), use K-means to get smaller sized clusters and then run more expensive algorithms on these smaller clusters.

12.6 Hierarchical

Hierarchical clustering is one of the simplest and widely used clustering techniques for analysis of gene expression data. The method follows an agglomerative approach, where the most similar expression profiles are joined together to form a group. These are further joined in a tree structure, until all data forms a single group. The dendrogram is the most intuitive view of the results of this clustering method.

There are several important parameters, which control the order of merging rows and sub-clusters in the dendrogram. The most important of these

is the linkage rule. After two most similar rows (clusters) are clubbed together, this group is treated as a single entity and its distances from the remaining groups (or rows) have to the re-calculated. **ArrayAssist** gives an option of the following linkage rules on the basis of which two clusters are joined together:

Complete Linkage Distance between two clusters is the greatest distance between the members of the two clusters

Single Linkage Distance between two clusters is the minimum distance between the members of the two clusters.

Average Linkage Distance between two clusters is the average of the pairwise distance between rows in the two clusters.

Centroid Linkage Distance between two clusters is the average distance between their respective centroids.

Median Linkage Distance between two clusters is the median of the pairwise distances between the rows in the two clusters.

Ward's Method This method is based on the ANOVA approach. It computes the sum of squared errors around the mean for each cluster. Then, two clusters are joined so as to minimize the increase in error.

Hierarchical clustering can be invoked by clicking on Clustering and selecting Hierarchical. Clustering will be carried out on the current dataset in the Spreadsheet. The *Parameters* dialog box will appear. Various clustering parameters to be set are as follows:

Cluster On Dropdown menu gives a choice of Rows, or Columns, or Both rows and columns, on which clusters can be formed. The default is Rows.

Distance Metric Dropdown menu gives seven choices; Euclidean, Squared Euclidean, Manhattan, Chebychev, Differential, Pearson Absolute and Pearson Centered. The default is Euclidean.

Linkage Rule The dropdown menu gives the following choices; complete, single, average, centroid, median, and wards. The default is complete.

Views The graphical views available with Hierarchical clustering are

- Dendrogram View

- Similarity Image View

Results of clustering will appear in the desktop, with each view as a separate window. Hierarchical and its output views will be added to the navigator.

Advantages and Disadvantages of Hierarchical Clustering: Hierarchical clustering builds a full relationship tree and thus gives a lot more relationship information than K-Means. However, it tends to connect together clusters in a local manner and therefore, small errors in cluster assignment in the early stages of the algorithm can be drastically amplified in the final result. Also, it does not output clusters directly; these have to be obtained manually from the tree.

12.7 Self Organizing Maps (SOM)

SOM Clustering is similar to K-means clustering in that it is based on a divisive approach where the input rows are partitioned into a fixed user defined number of clusters. Besides clusters, SOM produces additional information about the affinity or similarity between the clusters themselves by arranging them on a 2D rectangular or hexagonal grid. Similar clusters are neighbors in the grid, and dissimilar clusters are placed far apart in the grid.

The algorithm starts by assigning a random reference vector for each node in the grid. A gene is assigned to a node, called the winning node, on this grid based on the similarity of its reference vector and the expression vector of the gene. When a gene is assigned to a node, the reference vector is adjusted to become more similar to the assigned gene. The reference vectors of the neighboring nodes are also adjusted similarly, but to a lesser extent. This process is repeated iteratively to achieve convergence, where no gene changes its winning node. Thus, rows with similar expression vectors get assigned to partitions that are physically closer on the grid, thereby producing a topology that preserves the mapping from input space onto the grid.

In addition to producing a fixed number of clusters as specified by the grid dimensions, these proto-clusters (nodes in the grid) can be clustered further using hierarchical clustering, to produce a dendrogram based on the proximity of the reference vectors.

SOM clustering can be invoked by clicking on Clustering and selecting SOM. Clustering will be carried out on the current dataset in the Spreadsheet. The *Parameters* dialog box will appear. Various clustering parameters to be set are as follows:

Grid Topology This determines whether the 2D grid is hexagonal or rectangular. Choose from the dropdown list. Default topology is hexagonal.

Number of grid rows Specifies the number of rows in the grid. This value should be a positive integer. The default value is 3.

Number of grid columns Specifies the number of columns in the grid. This value should be a positive integer. The default value is 4.

Initial learning rate This defines the learning rate at the start of the iterations. It determines the extent of adjustment of the reference vectors. This decreases monotonically to zero with each iteration. The default value is 0.03.

Neighborhood type This determines the extent of the neighborhood. Only nodes lying in the neighborhood are updated when a gene is assigned to a winning node. The dropdown list gives two choices - Bubble or Gaussian. A Bubble neighborhood defines a fixed circular area, whereas a Gaussian neighborhood defines an infinite extent. However, the update adjustment decreases exponentially as a function of distance from the winning node. Default type is Bubble.

Initial neighborhood radius This defines the neighborhood extent at the start of the iterations. This radius decreases monotonically to 1 with each iteration. The default value is 5.

Number of iterations This is the upper bound on the maximum number of iterations. The default value is 50.

Run Batch SOM Batch SOM runs a faster simpler version of SOM when enabled. This is useful in getting quick results for an overview, and then normal SOM can be run with the same parameters for better results. Default is off.

Views The graphical views available with SOM clustering are

- U-Matrix
- Cluster Set View
- Dendrogram View
- Similarity Image View

Results of clustering will appear in the desktop, with each view as a separate window. SOM and its output views will be added to the navigator.

12.8 Eigen Value Clustering

Eigen Value clustering is based on the principle that Eigen vectors of the similarity matrix associated with the given set of rows contain information on how the rows cluster. The algorithm computes and processes these Eigen vectors to identify clusters one at a time. Each round of the algorithm permutes the rows based on the Eigen vectors obtained in such a way that one cluster automatically rises to the top. This cluster is removed and the process repeated. The time taken by this process depends upon the number of clusters there are in the data.

Eigen Value clustering can be invoked by clicking on Clustering and selecting Eigen Value. Clustering will be carried out on the current dataset in the Spreadsheet. The *Parameters* dialog box will appear. Various clustering parameters to be set are as follows:

Cluster On Dropdown menu gives a choice of Rows, or Columns, or Both rows and columns, on which clusters can be formed. The default is Rows.

Distance Metric This is the only clustering algorithm that gives the choice of the Angular distance metric. It is the default setting. Other choices in the dropdown list are; Euclidean, Squared Euclidean, Manhattan, Chebychev, Differential, Pearson Absolute and Pearson Centered.

Cutoff Ratio This defines a cut off for isolating the cluster which rises to the top. A larger value imposes a more aggressive cutoff. A value of 0 would give just one large cluster, and the number of clusters increases as this cutoff is increased. The default is 0.9.

Views The graphical views available with Eigen Value Clustering are:

- Cluster Set View
- Dendrogram View
- Similarity Image View

Results of clustering will appear in the desktop, with each view as a separate window. Eigen and its output views will be added to the navigator.

Advantages and Disadvantages of Eigen Value Clustering: Eigen Value Clustering produces permuted clusters, i.e., the order in which rows appear gives some indication of their relatedness (consecutive rows in a permutation are closer than far away rows). It is best at identifying large (as

a fraction of the total number of rows) coarse clusters. Smaller clusters can be identified by drilling down within a cluster and re-running the algorithm.

12.9 PCA Clustering

Principal Components Analysis (PCA) clustering finds principal components (i.e. Eigen vectors of the similarity matrix of the rows) and projects each gene to the nearest principal component. All rows associated with the same principal component in this way comprise a cluster.

PCA clustering can be invoked by clicking on Clustering and selecting PCA. Clustering will be carried out on the current dataset in the Spreadsheet. The *Parameters* dialog box will appear. Various clustering parameters to be set are as follows:

Cluster On Dropdown menu gives a choice of Rows, or Columns, or Both rows and columns, on which clusters can be formed. Default is Rows.

Number of Clusters This is the number of clusters desired finally. It cannot be greater than the number of principal components, which itself is at most the number of rows or number of columns, whichever is smaller.

Normalization Checking this option will normalize each column to mean 0 and variance 1 before the algorithm is run.

Views The graphical views available with PCA clustering are

- Cluster Set View
- Dendrogram
- Similarity Image View

Results of clustering will appear in the desktop, with each view as a separate window. PCA and its output views will be added to the navigator

Advantages and Disadvantages of PCA Clustering: PCA clustering is fast and can handle large datasets. Like K-means, it can be used to cluster a large dataset into coarse clusters which can then be clustered further using other algorithms. However, it does not provide a choice of distance functions. Further, the number of clusters it finds is bounded by the smaller of the number of rows and number of columns.

12.10 Random Walk

This clustering method is based on deterministic analysis of random walks on the weighted graph associated with a dataset [**?**]. A graph is a collection of points along with some edges joining pairs of points. If edges of the graph are assigned values called weights then it becomes a weighted graph. We construct the weighted graph as follows. Points in the graph are the samples. Each sample in the data set has a set of values which we use as co-ordinates for the corresponding point. Using the given distance measure we compute the nearest neighbors for that point. The number of nearest neighbors we compute is given by number of neighbors given as an input parameter. We now join each point to its nearest neighbors with edges that are weighted. The weights are computed as the inverse of the distance between two neighboring samples. Thus nearer neighbors receive a higher weight than farther neighbors. In this way similar rows receive a higher weight than dissimilar ones. The algorithm then performs a 'sharpening' pass which is repeated up to the number of iterations specified in the input parameter list. This sharpening pass is based on a random walk from a sample along the edges that connect to it for a distance of the walking depth. This further differentiates the similar from the dissimilar rows. Due to sharpening the edges within a group of points which ought to be together (in a cluster) become stronger and edges across clusters weaken. Using these sharpened weights we construct a dendrogram using the linkage rule specified in the input parameter list.

Random Walk clustering can be invoked by clicking on Clustering and selecting RandomWalk. Clustering will be carried out on the current dataset in the Spreadsheet. The *Parameters* dialog box will appear. Various clustering parameters to be set are as follows:

Cluster On Dropdown menu gives a choice of Rows, or Columns, or Both rows and columns, on which clusters can be formed. Default is Rows.

Distance Metric Choices in the dropdown list are; Euclidean, Squared Euclidean, Manhattan, Chebychev, Differential, Pearson Absolute and Pearson Centered. The default metric is Euclidean.

Linkage Rule Choices in the dropdown list are; Average, Complete and Single. The default metric is Average. Single Linkage is good for dense datasets but it produces lot of outliers. Complete Linkage has the disadvantage of breaking up clusters into unnatural ones. It is

advisable to try all three linkage rules and then choose the best among them.

Walking Depth This determines the length of the random walk performed. The default value is 3. Increasing this quantity will increase the running time substantially. Further, increasing it too much dilutes the clustering quality. Typically a depth of walk between 3 - 6 is enough to produce quality results.

Number of Iterations This controls the number of sharpening passes done for weight adjustment. The default is 2 iterations. In general 1 or 2 iterations are enough for good clustering.

Number of Neighbors This is the probably the most crucial parameter that determines the clustering quality. The default value is 30. For dense data sets it is better to go for higher values like 40-50. For sparse datasets, about 20 neighbors is reasonable.

Views The graphical views available with RandomWalk clustering are:

- Dendrogram View
- Similarity Image View

Results of clustering will appear in the desktop, with each view as a separate window. RandomWalk and its output views will be added to the navigator.

Advantages and Disadvantages of Random Walk Random Walk clustering,when used without selecting the similarity image, requires little memory and it can be used for datasets upto 20,000 rows on a 256MB RAM machine. The disadvantage with this algorithm is that the results are highly sensitive to the input parameter list, especially on Linkage Rule and the number of neighbors. It is therefore best to test this with all possible combinations of input parameters.

12.11 Guidelines for Clustering Operations

12.11.1 How to Identify k in K-Means Clustering

The K-Means algorithm requires a user-defined value of k for execution. This value may be available in certain cases, for example, number of treatments, number of patient groups, etc. Principal Component Analysis (PCA) results

can also be used to determine the value of k by visually estimating the number of clusters in the projections along the principal components. It is possible to run Hierarchical clustering first to get an overall idea of the number of clusters, and seed K-Means with this value. Finally, the similarity image view can also be used to identify the number of clusters in the data. Use any clustering algorithm and look at the similarity view (this option cannot be used on very large datasets as it is memory intensive, see below for some figures). The number of high intensity blocks along the diagonal in this view is the number of clusters in the data, adjusting for split clusters as described earlier in Similarity Image section.

12.11.2 What is a Recommended Sequence for using Algorithms

The choice of clustering algorithm is driven by several factors, including the size of the dataset, nature of data and any a priori information about the data. Ideally, it is recommended that several of these be tried to evaluate the consistency of results and determine which one works the best for a given dataset.

The table below depicts a comparison of these techniques with their tradeoffs. These times were measured on a 1.6GHz Pentium machine with 1.5MB RAM. All datasets used had 133 rows. Note that K-Means, SOM, PCA and Random Walk can be run for 20,000 rows without the Similarity Image option on a 256MB RAM machine. Hierarchical clustering can run with up to 8000 rows on a 256MB RAM machine and 20,000 rows on a 2MB RAM machine.

Algorithm	5000 rows	10000 rows	20000 rows
K-Means	0m:01s	0m:01s	0m:05s
Hierarchical	0m:17s	1m:16s	4m:02s
SOM	0m:31s	1m:01s	3m:02s
Eigen Value	0m:55s	3m:43s	44m:21s
Random Walk	0m:13s	0m:55s	3m:00s
PCA	0m:12s	0m:24s	0m:49s

Chapter 13

Classification: Learning and Predicting Outcomes

13.1 What is Classification

Classification algorithms in **ArrayAssist** are a set of powerful tools that allow researchers to exploit microarray data for learning-based prediction of outcomes of gene expression. These tools stretch the use of microarray technology into the arena of diagnostics and understanding the genetic basis of complex diseases. In **ArrayAssist**, classification comprises a set of supervised learning algorithms, which construct a model from a training dataset in which the separation of genes into classes has already been done. This model is then used to predict classes for new unclassified data.

Typically, classification algorithms can be applied to microarray data in two ways. The first type works at the level of individual genes. For example, if expression profiles as well as function information are available for a collection of genes, then this information can be used to learn a model which can then predict functions for new genes given their expression profiles alone. The second type works at the level of experiments or samples. For example, given gene expression data for different kinds of cancer samples, a model which can predict the cancer type for an new sample can be learnt from this data.

Model building for classification in **ArrayAssist** is done using four powerful machine learning algorithms - Decision Tree (DT), Neural Network (NN), Support Vector Machine (SVM), and Models built with these algorithms can then be used to classify samples or genes into discrete classes.

In addition, a Linear Multivariate Regression algorithm allows for pre-

diction of continuous variables like survival indices. Look at the *Linear Multivariate Regression* chapter for details.

The models built by these algorithms range from visually intuitive (as with Decision Trees) to very abstract (as for Support Vector Machines). Further, the classification algorithms vary in their ability to handle multiple classes (SVM can distinguish between two classes only while the others can handle multiple classes) and discrete variables (only axis parallel DT can handle discrete variables, e.g., tumor samples may be marked as large, small or medium and this may be one of the factors in learning a model). Together, these methods constitute a comprehensive toolset for learning, classification and prediction.

13.2 Classification Pipeline Overview

13.2.1 Dataset Orientation

All classification and prediction algorithms in **ArrayAssist** predict classes/values for rows in the dataset. Therefore, when predicting gene function classes, genes should be along rows and samples/experiments along columns. And when predicting phenotypic properties of samples based on gene expression, samples should be along rows and genes should be along columns. To get the right orientation, use the transpose feature available from the Data menu on the main menu bar if necessary. This will create a new dataset in a new datatab that can be using for classification.

13.2.2 Class Labels and Training:

The next step, to learn a model from the data in the spreadsheet, Training needs to be performed using one of the algorithms available. For training, each row needs to have an associated Class Label which describes the class or the value of the phenotypic variable associated with the row. For example, if genes are being classified based on function then the functional class of each gene needs to be specified. And if samples are being classified based on tumor categories, then the tumor category of each sample needs to be specified. Finally, if what is being predicted is a phenotypic variable, e.g. a survival index, for a sample, then the value of this variable needs to be specified for each sample. These values must appear in a special column which contains the Class Labels. This column can be specified before execution by specifying the appropriate column in the *Columns* section of Algorithm Parameters dialog. This is a frequently needed operation, and the Class

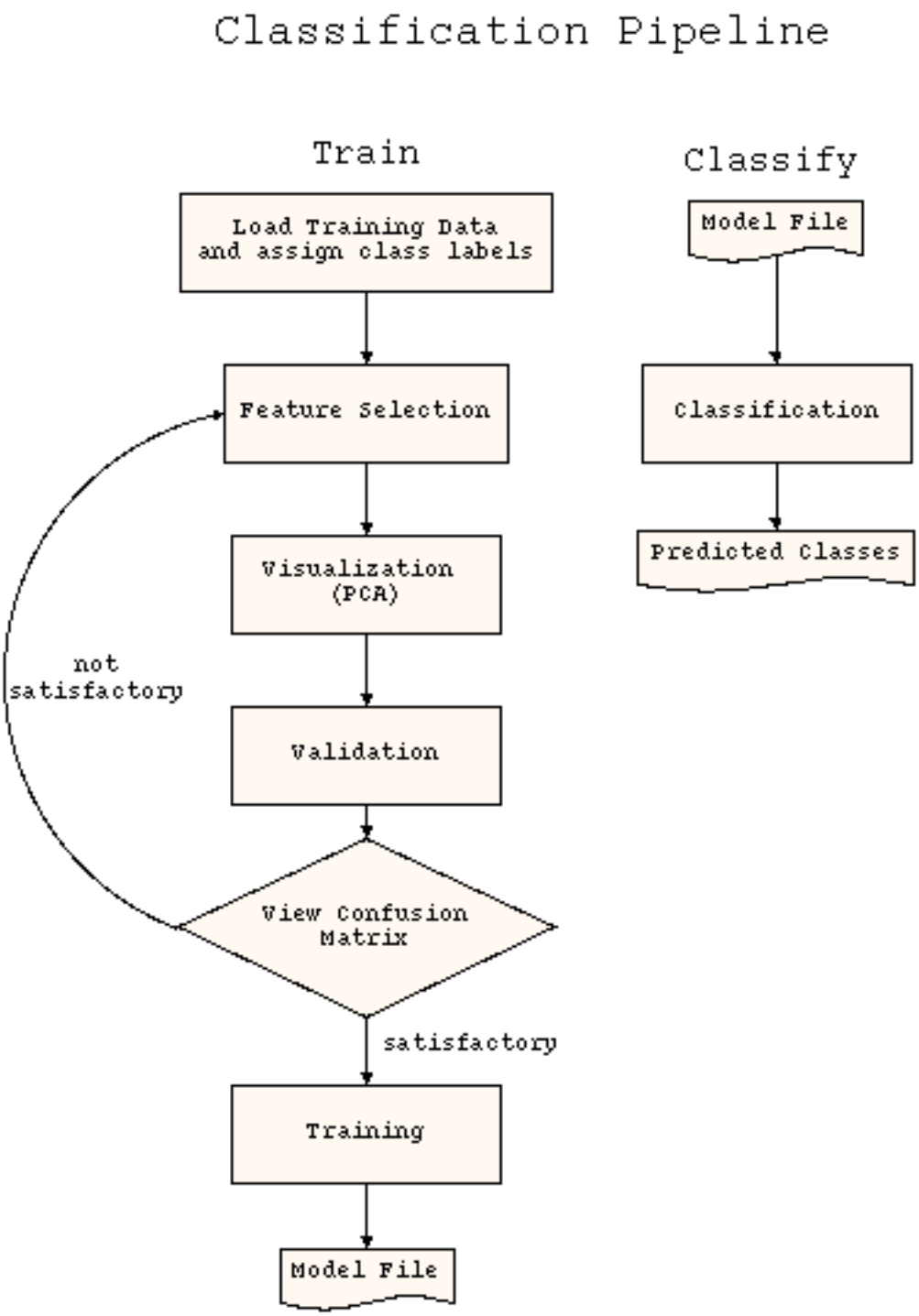


Figure 13.1: Classification Pipeline

Label column is used in several other visualizations as well; so a convenient way is provided to permanently mark a column as a Class Label column in the dataset. See the Creating a Class Label column heading below to see how existing columns can be marked as Class Label columns, or how a new Class Label column is created.

Once the Class Label column is set up, training can be run using one of the several learning algorithms available in **ArrayAssist**. This process will mine the data and come up with a model which can be saved in a file for future use. The actual meaning and representation of this model varies with the method used. Decision trees output models in which sequences of decisions of the following form are represented as trees - if gene X has expression value less than A and gene Y has expression value more than B then the associated sample is cancerous. Neural Networks and Support Vector Machines output models which are more abstract. The training process also comes up with a predicted class or variable value for each of the rows as predicted by the model being constructed. These predictions give some feel for how good the model is. However, it is dangerous to trust models based on these predictions as the training process often has a tendency to over-fit, i.e. yield models which memorize the data. If this is indeed the case then these models will not work well in the Classification stage, i.e. when predicting on new data with unknown Class Labels.

13.2.3 Feature Selection:

Very often, model prediction accuracies and algorithm speeds can be substantially increased by performing training not with the whole feature set but with only a subset of relevant and important features. Several tests for selecting important features are available in **ArrayAssist**. Once the dataset is restricted to these features, this feature set needs to be validated, as above.

Features and Validation:

To give a feel for how well a model obtained in the training step would do in the classification step on a new dataset, we need to run Validate on the feature set. The feature set is the set of columns in the dataset. For example, if samples are being classified into tumor categories then each column would represent a gene and classification decisions would be based on expression values of some or all of these genes; in this case, the set of genes constitutes the feature set. The aim in validation is to check whether the given set

of features in the dataset is powerful enough to yield good models which can make accurate predictions on new datasets. In the absence of this new dataset, the existing dataset is split into two parts by the validation process - one part is used for training; the resulting model is applied on the second part, and the accuracies of the predictions are output. If these predictions are accurate, then the feature set is a good one and the model obtained in training is likely to perform well on new datasets, provided of course that the training dataset captures the distributional variations in these new datasets.

13.2.4 Classification:

If the validation accuracy obtained above is high then training can be used to build a model which will then be used for classification on new datasets. High validation accuracies indicate that this model is likely to work well in practice.

Note: All classification algorithms in **ArrayAssist** for prediction of discrete classes (i.e. SVM, NN, NB and DT) allow for validation, training and classification.

13.3 Specifying a Class Label Column

Training and validation require that all rows have Class Labels associated with them. The column containing the Class Labels can be specified before execution by specifying the appropriate column in the *Columns* section of Algorithm Parameters dialog. This is a frequently needed operation, and the Class Label column is used in several other visualizations as well; so a convenient way is provided to permanently mark a column as a Class Label column in the dataset.

Specifying a Class Label Column in the dataset An existing column can be permanently marked as the Class Label column in the dataset using the *Mark* command. Click the *Mark* icon in the spreadsheet toolbar (or select Data → Mark option) and specify an existing column as Class Label column. NOTE: Only columns with categorical values can be marked as Class Label columns. See *Data Properties* command for more information.

Creating a new Class Label Column If a Class Label column does not already exist in the dataset, then there are multiple ways to create a new Class Label column.

- Use the *Create New Column Using Formula* command to append a new column to the dataset with the appropriate values. This command is accessible from the *Create New Column* icon in the spreadsheet toolbar, as well as Data → Column Operations→ Create New Column menu item.
- Select rows corresponding to a class (either via the lasso, or from the spreadsheet). Use the *Data → Row Operations → Label As* command to assign a Class Label of choice to the selected rows. If no Class Label column exists, a new String column is appended to the dataset, and the Class Label value is set to user-specified value for the selected rows. If a Class Label column already exists, then the values in the selected rows are overridden with user-specified value. This operation requires that the dataset be unlocked.

Directly Edit values in the dataset via the spreadsheet by editing appropriate cells in the table.

13.4 Viewing Data for Classification

13.4.1 Viewing Data using Scatter Plots and Matrix Plots

ArrayAssist provides tools to visualize the data to be classified. If a Class Label column is marked on the spreadsheet, all scatter plots and the matrix plot will show each class in a different color. Inspection of scatter plots can provide pointers to appropriate classification models. For example, if the scatter plot shows adequate separation of classes then, Decision Trees, a linear SVM or Neural Nets with no hidden layers may be appropriate for a classification model. However, if the data were intermixed, a higher kernel order function for SVM or a Naive Bayesian classification model may be more effective.

The following tools can be used to view spreadsheet data for classification:

Scatter Plot: Class separation can be visualized by either coloring based on Class Label column, or choosing shapes based on Class Label column.

Matrix Plot: Class separation can be visualized by coloring based on Class Label column. The Matrix plot of the selected columns shows all

pairwise two-way plots. These can be examined for separability of classes across columns, and then the axes along which the classes are best separated can be chosen for further analysis.

13.5 Feature Selection

The next step in classification analysis is to select those features in the dataset that would help classify the data. Visualizing the data with PCA gives insight into the existing level of separation. If it is not satisfactory enough to proceed to the learning algorithms, feature selection techniques can be tried. For example, when gene expression data across experiments has redundant information, a subset of experiments containing important information can be selected for analysis from the original dataset, to classify genes most effectively. Similarly, if experiments are being classified, the genes contributing the maximum information can be selected. This is called feature selection. A classification model learnt with too many features may over-fit the model for the training data, and may not be generalizable to classifying new data satisfactorily. Good feature selection also improves the speed and accuracy of learning algorithms.

ArrayAssist has statistical tools to help select important features for classification and reduce the dimensionality of the data. These tests are done on all features (i.e. columns of data) with Class Labels used to group rows together. Statistical tests of hypothesis check to see which features show significant variation across groups and produce an associated significance or p-value for each feature. A chosen number of best features can be obtained by cutting off based on an appropriate choice of p-value.

13.5.1 ANOVA

ANOVA performs a parametric test to check whether the means of two or more classes within a column are equal, assuming that each group within a column comes from a normal distribution. Visualizing the distribution of all columns using Descriptive Statistics will give a rough indication of this information. If the distribution is not normal, the non-parametric Kruskal-Wallis test may be more appropriate.

To perform ANOVA: In the Classification dropdown menu, select Feature Selection, and choose ANOVA. In the ANOVA dialog box, select whether variances are to be Equal or Unequal from the dropdown list.

If there is reason to believe that the variance or spread of the distribution for the two classes will be different, the Unequal option should be chosen. The default is Equal. Click OK to execute the command. The ANOVA results appear under the current spreadsheet in the navigator, along with its result window.

ANOVA is performed on every column of the spreadsheet The Sorted p-value table in the ANOVA - p-value window has three columns. The first column contains feature names sorted in descending order of p-value. The second column gives the respective F-statistics, and the third column gives the p-value.

Based on this analysis, features can be selected and saved to a file, or a new dataset can be created for further classification analysis. Features can be selected based on the p-value, or the rank of the p-value, as explained below.

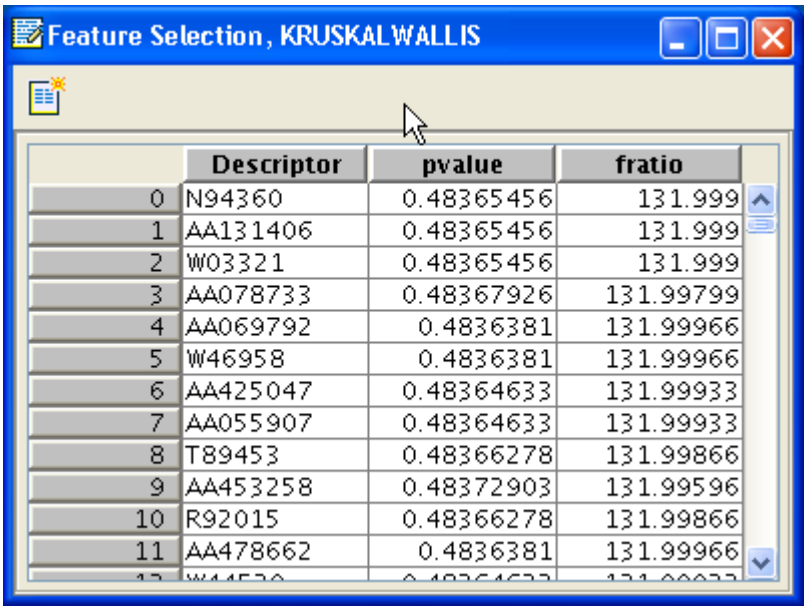
13.5.2 Kruskal-Wallis Test

Kruskal-Wallis is a non-parametric test of difference between distributions of two or more classes, when they cannot be assumed to have normal distributions. The test checks whether the distributions of various classes within a column are similar. If these are indeed different within a column, this feature could be a good feature for the classification model.

To perform the Kruskal-Wallis: In the Classification dropdown menu, select Feature Selection, and click on Kruskal-Wallis. Select the class label column and click OK to complete. The Kruskal-Wallis results appear under the current spreadsheet in the navigator, along with its result window.

The Kruskal-Wallis test is performed on every column of the spreadsheet The Sorted p-value table in the Kruskal-Wallis - p-value window has three columns. The first column contains features sorted in ascending order of p-value. The second column gives p-value, and the third column gives the respective Z-statistics.

Based on this analysis, features can be selected and saved to a file, or a new dataset can be created for further classification analysis. Features can be selected based on the p-value, or the rank of the p-value, as explained below.



	Descriptor	pvalue	fratio
0	N94360	0.48365456	131.999
1	AA131406	0.48365456	131.999
2	W03321	0.48365456	131.999
3	AA078733	0.48367926	131.99799
4	AA069792	0.4836381	131.99966
5	W46958	0.4836381	131.99966
6	AA425047	0.48364633	131.99933
7	AA055907	0.48364633	131.99933
8	T89453	0.48366278	131.99866
9	AA453258	0.48372903	131.99596
10	R92015	0.48366278	131.99866
11	AA478662	0.4836381	131.99966

Figure 13.2: Feature Selection Output

13.5.3 Saving Features and Creating New Datasets

Having performed one of the above two statistical tests, the results can be saved or applied to create a new dataset with columns restricted to the selected features.

Click *Save Feature File* or *Create New Dataset* in the window toolbar. In the Select Features dialog box use the Select dropdown menu to choose whether All features, those Based on p-value or those Based on rank are to be selected. Even if you use Create New Dataset directly, it is also advisable to save the features to a file for later invocation at the time of classification.

Selecting features based on p-value If features Based on p-value are to be selected, then enter the required p-value in the p-value field. The default is 0.05. This implies that the hypothesis of unequal means is accepted at a p-value of 0.05 and the means of the two distributions are considered different.

Selecting features based on Rank If features are to be selected based on the ranking in p-value, then give the number of features to be selected from the Top of the p-value ranking, say the top 20 features.

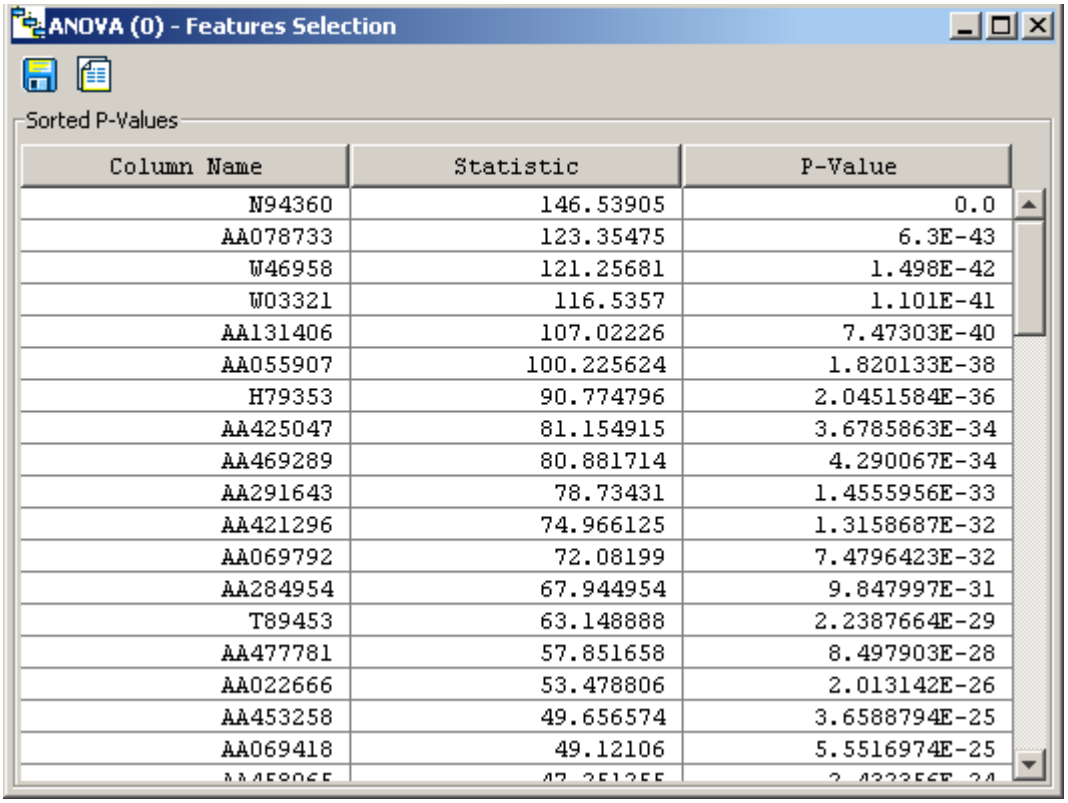


Figure 13.3: Feature Selection Output

Saving features or Creating new Dataset In the Save dialog box, give the name of the file with an .fts extension in which features are to be saved. Click Save to complete. Alternatively, if the Create New Dataset option was chosen then give the name of the new dataset. The current spreadsheet restricted to the chosen features will appear on a new spreadsheet, along with the identifier and Class Label columns.

13.5.4 Feature Selection from File

Suppose, after visualizing the data for classification, and running the statistical tests for feature selection, the selected features have been written to a file. Then, feature selection from file can be used to create a new dataset with the selected features, for further use in training a model, or for classifying an unknown dataset with a previously learned model.

Feature Selection from File In the Classification dropdown menu, select Feature Selection and choose the File-based Feature Selection option.

Choose .fts file In the *Parameters* dialog box, Browse the required file using the *Open* dialog box. The file from which features are to be selected must have the extension .fts.

Note: A dataset created by feature selection from a file will have only the data columns for selected features along with the columns that were marked as Identifier and Class Label in the parent dataset. It will not contain any other string or data columns. If a model is constructed from a dataset obtained from a larger dataset using feature selection, and this model needs to be applied on a new dataset for prediction of unknown Class Labels, then feature selection from file will need to be run on this new dataset; classification will work only on the resulting feature selected dataset. Therefore, it is advisable to save features to a file whenever feature selection is performed.

13.6 The Three Steps in Classification

Classification is an interactive process where microarray data is visualized, appropriate features are selected, and then a classification model is built. **ArrayAssist** has four classification algorithms - Decision Tree (Axis Parallel and Oblique). Neural Network, Support Vector Machine (SVM), , and each of these can be used with a variety of parameters. Building a classification model for microarray data involves experimenting with different algorithms

and parameters. Visualization of classified data gives clues to the most suitable model to be chosen. For example, if the scatter plots and PCA visualization reveal a good separation of data, the SVM linear classifiers or Decision Trees may be reasonable models. On the other hand, if the classes are intermixed in the scatter plots and PCA, then nonlinear classifiers like Neural Nets or SVMs with higher order kernels may be more appropriate. Naive Bayesian classifier is a parametric classifier and works best when data is normally distributed along each axis.

Classification in **ArrayAssist** has three components - Train, Validate and Classify. Training involves using a dataset with known class values, and learning a model from that dataset. However, models that fit the training dataset very well may mis-classify new data points. Such over-fitting of the training data will most likely yield a model that cannot be generalized and, therefore, would not be useful. Therefore, an algorithm and its associated parameters must be validated before they are used to classify new data. This process involves segmenting the training data into two sets. One set is used for training and the other for testing the model. Typically, validation should be done with a variety of algorithms and model parameters, and results monitored to choose the best combination. This combination can then be used to build a model with the entire training dataset, and then to classify new data.

13.6.1 Validate

Validation helps to choose the right set of features, an appropriate algorithm and associated parameters for a particular dataset. Validation is also an important tool to avoid over-fitting models on training data as over-fitting will give low accuracy on validation. Validation can be run on the same dataset using various algorithms and altering the parameters of each algorithm. The results of validation, presented in the Confusion Matrix (a matrix which gives the accuracy of prediction of each class), are examined to choose the best algorithm and parameters for the classification model.

Two types of validation have been implemented in **ArrayAssist**.

Leave One Out: All data with the exception of one row is used to train the learning algorithm. The model thus learnt is used to classify the remaining row. The process is repeated for every row in the dataset and a Confusion Matrix is generated.

N-fold: The rows in the input data are randomly divided into N equal parts; N-1 parts are used for training, and the remaining one part is

used for testing. The process repeats N times, with a different part being used for testing in every iteration. Thus each row is used at least once in training and once in testing, and a Confusion Matrix is generated. This whole process can then be repeated as many times as specified by the number of repeats.

The default values of three-fold validation and one repeat should suffice for most approximate analysis. If greater confidence in the classification model is desired, the Confusion Matrix of a 10-fold validation with three repeats needs to be examined. However, such trials would run the classification algorithm 30 times and may require considerable computing time with large datasets.

13.6.2 Train

Each of the learning algorithms in **ArrayAssist** can be trained with a (hopefully representative) dataset that has Class Labels. The results of training yield a Model, a Report, a Confusion Matrix and a plot of the Lorenz Curve. These views will be described in detail later.

13.6.3 Classify

Once the learning algorithm has been trained and a model fit is available, it can be used to classify new data. For example, if Neural Net has been used develop the model, then only Neural Net can be used to classify. The results are presented in a Report with newly assigned Class Labels. If Class Labels are already present in the input dataset, a Confusion Matrix and the Lorenz Curve are also reported.

13.7 Decision Trees

A Decision Tree is best illustrated by an example. Consider three samples belonging to classes A,B,C, respectively, which need to be classified, and suppose the rows corresponding to these samples have values shown below:

Then the following sequence of Decisions classifies the samples - if feature 1 is at least 4 then the sample is of type A, and otherwise, if feature 2 is bigger than 10 then the sample is of Type B and if feature 2 is smaller than 10 then the sample is of type C. This sequence of if-then-otherwise decisions can be arranged as a tree. This tree is called a decision tree.

	Feature 1	Feature 2	Feature 3	Class Label
Sample 1	4	6	7	A
Sample 2	0	12	9	B
Sample 3	0	5	7	C

Table 13.1: Decision Tree Table

ArrayAssist implements two types of Decision Trees - Axis Parallel and Oblique. In an axis parallel tree, decisions at each step are made using one single feature of the many features present, e.g. a decision of the form if feature 2 is less than 10. In contrast, in oblique decision trees, decisions at each step could be made using linear combinations of features, e.g. if 3 times feature 2 plus 4 times feature 5 is less than 10.

The decision points in a decision tree are called internal nodes. A sample gets classified by following the appropriate path down the decision tree. All samples which follow the same path down the tree are said to be at the same leaf. The tree building process continues until each leaf has purity above a certain specified threshold, i.e., of all samples which are associated with this leaf, at least a certain fraction comes from one class. Once the tree building process is done, a pruning process is used to prune off portions of the tree to reduce chances of over-fitting.

Axis parallel decision trees can handle multiple class problems. Both varieties of decision trees produce intuitively appealing and visualizable classifiers.

The following sections give Decision Tree parameters for training, validation and classification.

13.7.1 Decision Tree Train

To train a Decision Tree, select *Training* from the Classification menu and choose the Decision Tree. The *Parameters* dialog box for Decision Tree will appear. The training input parameters to be specified are as follows:

Decision Tree Type One of two types of Decision Trees can be selected from the dropdown menu - Axis Parallel and Oblique. The default is Axis Parallel.

Pruning Method The options available in the dropdown menu are - Minimum Error, Pessimistic Error, and No Pruning. The default is Minimum Error. The No Pruning option will improve accuracy at the cost of potential over-fitting.

Goodness Function Two functions are available from the dropdown menu - Gini Function and Information Gain. This is implemented only for the Axis Parallel decision trees. The default is Gini Function.

Allowable Leaf Impurity Percentage (Global or Local) If this number is chosen to be x with the global option and the total number of rows is y, then tree building stops with each leaf having at most x*y/100 rows of a class different from the majority class for that leaf. And if this number is chosen to be x with the local option, then tree building stops with at most $x\%$ of the rows in each leaf having a class different from the majority class for that leaf. The default value is 1% and Global. Decreasing this number will improve accuracy at the cost of over-fitting.

Number of Iterations Specify the number of iterations. This parameter is used only for the oblique decision tree. The default value is 1000.

Learning Rate This parameter is also used only for the oblique decision tree. The default is 0.1.

The results of training with Decision Tree are displayed in the navigator. The Decision Tree view appears under the current spreadsheet and the results of training are listed under it. These consist of the Decision Tree model with parameters which can be saved as an .mdl file, a Report, a Confusion Matrix, and a Lorenz Curve, all of which will be described later.

13.7.2 Decision Tree Validate

To validate, select *Validation* from the Classification dropdown menu and choose the decision tree. The *Parameters* dialog box for Decision Tree Validation will appear. In addition to the parameters explained above for Decision Tree training, the following validation specific parameters need to be specified.

Validation Type Choose one of the two types from the dropdown menu - Leave One Out, N-Fold. The default is Leave One Out.

Number of Folds If N-Fold is chosen , specify the number of folds. The default is 3.

Number of Repeats The default is 1.

The results of validation with Decision Trees are displayed in the navigator. The Decision Tree view appears under the current spreadsheet and the results of validation are listed under it. They consist of the Confusion Matrix and the Lorenz Curve. The Confusion Matrix displays the parameters used for validation. If the validations results are good these parameters can be used for training.

13.8 Neural Network

Neural Networks can handle multi-class problems, where there are more than two classes in the data. The Neural Network implementation in **ArrayAssist** is the multi-layer perceptron trained using the back-propagation algorithm. It consists of layers of neurons. The first is called the input layer and features for a row to be classified are fed into this layer. The last is the output layer which has an output node for each class in the dataset. Each neuron in an intermediate layer is interconnected with all the neurons in the adjacent layers.

The strength of the interconnections between adjacent layers is given by a set of weights which are continuously modified during the training stage using an iterative process. The rate of modification is determined by a constant called the learning rate. The certainty of convergence improves as the learning rate becomes smaller. However, the time taken for convergence typically increases when this happens. The momentum rate determines the effect of weight modification due to the previous iteration on the weight modification in the current iteration. It can be used to help avoid local minima to some extent. However, very large momentum rates can also push the neural network away from convergence.

The performance of the neural network also depends to a large extent on the number of hidden layers (the layers in between the input and output layers) and the number of neurons in the hidden layers. Neural networks which use linear functions do not need any hidden layers. Nonlinear functions need at least one hidden layer. There is no clear rule to determine the number of hidden layers or the number of neurons in each hidden layer. Having too many hidden layers may affect the rate of convergence adversely. Too many neurons in the hidden layer may lead to over-fitting, while with too few neurons the network may not learn.

The following sections give Neural Network parameters for training, validation and classification.

13.8.1 Neural Network Train

To train a Neural Network, select *Training* from the Classification menu and choose *Neural Network*. The *Parameters* dialog box for Neural Network will appear. The training input parameters to be specified are as follows:

Number of Layers Specify the number of hidden layers, from layer 0 to layer 9. The default is layer 0, i.e., no hidden layers. In this case, the Neural Network behaves like a linear classifier.

Set Neurons This specifies the number of neurons in each layer. The default is 3 neurons. Vary this parameter along with the number of layers.

Starting with the default, increase the number of hidden layers and the number of neurons in each layer. This would yield better training accuracies, but the validation accuracy may start falling after an initial increase. Choose an optimal number of layers, which yield the best validation accuracy. Normally, up to 3 hidden layers are sufficient. A typical configuration would be 3 hidden layers with 7,5,3 neurons, respectively.

Number of Iterations The default is 100 iterations. This is normally adequate for convergence.

Learning Rate The default is a learning rate of 0.7. Decreasing this would improve chances of convergence but increase time for convergence.

Momentum The default is a 0.3.

The results of training with Neural Network are displayed in the navigator. The Neural Network view appears under the current spreadsheet and the results of training are listed under it. They consist of the Neural Network model with parameters which can be saved as an .mdl file, a Report, a Confusion Matrix, and a Lorenz Curve, all of which will be described later.

13.8.2 Neural Network Validate

To validate, select *Validation* from the Classification dropdown menu and choose Neural Network. The *Parameters* dialog box for Neural Network Validation will appear. In addition to the parameters explained above for Neural Network training, the following validation specific parameters need to be specified.

Validation Type Choose one of the two types from the dropdown menu - Leave One Out, N-Fold. The default is Leave One Out.

Number of Folds If N-Fold is chosen, specify the number of folds. The default is 3.

Number of Repeats The default is 1.

The results of validation with Neural Network are displayed in the navigator. The Neural Network view appears under the current spreadsheet and the results of validation are listed under it. They consist of the Confusion Matrix and the Lorenz Curve. The Confusion Matrix displays the parameters used for validation. If the validations results are good these parameters can be used for training.

13.9 Support Vector Machines

Support Vector Machines (SVM) is a binary classifier (i.e. it can be used only to classify between two groups). It attempts to separate rows into two classes by imagining these rows to be points in space and then determining a separating plane which separates the two classes of points. While there could be several such separating planes, the algorithm finds a good separator which maximizes the separation between the two classes of points. The power of SVMs stems from the fact that before this separating plane is determined, the points are transformed using a so called kernel function so that separation by planes post application of the kernel function actually corresponds to separation by more complicated surfaces on the original set of points. In other words, SVMs effectively separate point sets using non-linear functions and can therefore separate out intertwined sets of points.

The **ArrayAssist** implementation of SVMs, uses a unique and fast algorithm for convergence based on the Sequential Minimal Optimization method. It supports three types of kernel transformations - Linear, Polynomial and Gaussian. In all these kernel functions, it so turns out that only the dot product (or inner product) of the rows in important and that the rows themselves do not matter, and therefore the description of the kernel function choices below is in terms of dot products of rows, where the dot product between rows a and b is denoted by $x(a).x(b)$.

The Linear Kernel is represented by the inner product given by the equation $x(a).x(b)$.

The Polynomial Kernel is represented by a function of the inner product given by the equation $(k_1[x(a).x(b)]+k_2)^p$, where p is a positive integer.

The Gaussian Kernel is given by the equation $e^{-(\frac{x(a)-x(b)}{\sigma})^2}$

Polynomial and Gaussian kernels can separate intertwined datasets but at the risk of over-fitting. Linear kernels cannot separate intertwined datasets but are less prone to over-fitting and therefore, more generalizable.

An SVM model consists of a set of support vectors and associated weights called Lagrange Multipliers, along with a description of the kernel function parameters. Support vectors are those points which lie on (actually, very close to) the separating plane itself. Since small perturbations in the separating plane could cause these points to switch sides, the number of support vectors is an indication of the robustness of the model; the more this number, the less robust the model. The separating plane itself is expressible by combining support vectors using weights called Lagrange Multipliers.

For points which are not support vectors, the distance from the separating plane is a measure of the belongingness of the point to its appropriate class. When training is performed to build a model, these belongingness numbers are also output. The higher the belongingness for a point, the more the confidence in its classification.

The following sections give SVM parameters for training, validation and classification.

13.9.1 SVM Train

To train using the SVM method, in the Classification dropdown menu, select *Training* and choose *Support Vector Machine*. The *Parameters* dialog box for Support Vector Machine Training will appear. The training input parameters to be specified are as follows:

Kernel Type Available options in the dropdown menu are - Linear, Polynomial, and Gaussian. The default is Linear.

Max Number of Iterations A multiplier to the number of rows in the spreadsheet needs to be specified here. The default multiplier is 100. Increasing the number of iterations might improve convergence, but will take more time for computations. Typically, start with the default number of iterations and work upwards watching any changes in accuracy.

Cost This is the cost or penalty for misclassification. The default is 100. Increasing this parameter has the tendency to reduce the error in classification at the cost of generalization. More precisely, increasing this may lead to a completely different separating plane which has either more support vectors or less physical separation between classes but fewer misclassifications.

Ratio This is the ratio of the cost of misclassification for one class to the cost of the misclassification for the other class. The default ratio is 1.0. If this ratio is set to a value r, then the cost of misclassification for the class corresponding to the first row is set to the cost of misclassification specified in the previous paragraph, and the cost of misclassification for the other class is set to r times this value. Changing this ratio will penalize misclassification more for one class than the other. This is useful in situations where, for example, false positives can be tolerated while false negatives cannot. Then setting the ratio appropriately will have a tendency to control the number of false negatives at the expense of possibly increased false positives. This is also useful in situations where the two classes have very different sizes. In such situations, it may be useful to penalize classifications much more for the smaller class than the bigger class

Kernel Parameter (1) This is the first kernel parameter k1 for polynomial kernels and can be specified only when the polynomial kernel is chosen. Default if 0.1.

Kernel parameter (2) This is the second kernel parameter k2 for polynomial kernels. Default is set to 1. It is preferable to keep this parameter non-zero.

Exponent This is the exponent of the polynomial for a polynomial kernel (p). The default value is 2. A larger exponent increases the power of the separation plane to separate intertwined datasets at the expense of potential over-fitting.

Sigma This is a parameter for the Gaussian kernel. The default value is set to 1.0. Typically, there is an optimum value of sigma such that going below this value decreases both misclassification and generalization and going above this value increases misclassification. This optimum value of sigma should be close to the average nearest neighbor distance between points.

The results of training with SVM are displayed in the navigator. The Support Vector Machine view appears under the current spreadsheet and the results of training are listed under it. They consist of the SVM model which can be saved as an .mdl file, a Report, a Confusion Matrix, and a Lorenz Curve, all of which will be described later.

13.9.2 SVM Validate

To validate, select *Validation* from the Classification dropdown menu and choose *Support Vector Machine*. The *Parameters* dialog box for Support Vector Machine Validation will appear. In addition to the parameters explained above for SVM training, the following validation specific parameters need to be specified.

Validation Type Choose one of the two types from the dropdown menu - Leave One Out, N-Fold. The default is Leave One Out.

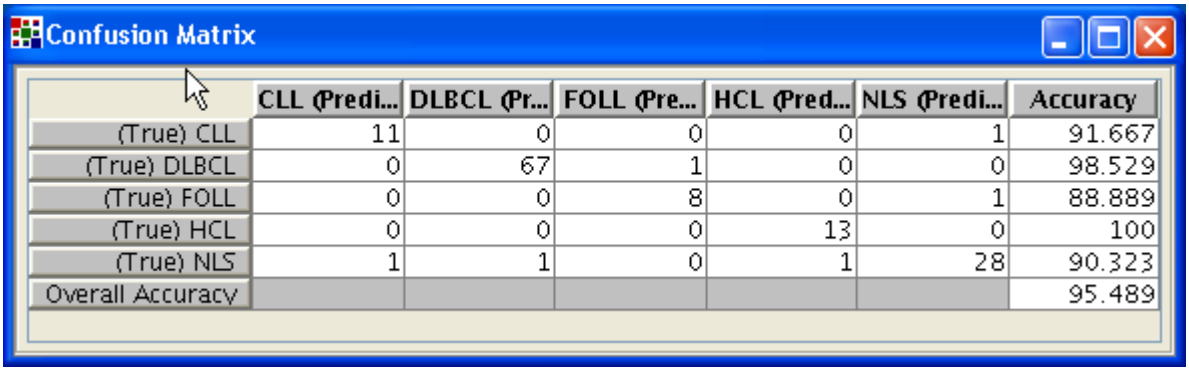
Number of Folds If N-Fold is chosen, specify the number of folds. The default is 3.

Number of Repeats The default is 1.

The results of validation with SVM are displayed in the navigator. The Support Vector Machine view appears under the current spreadsheet and the results of validation are listed under it. They consist of the Confusion Matrix and the Lorenz Curve. The Confusion Matrix displays the parameters used for validation. If the validations results are good then these parameters can be used for training.

13.10 Classification or Predicting Outcomes

To classify or predict the outcome of a new sample, a classification model must be already built and be available as a mdl file. To classify, from the Classification menu, choose Classify. The *Parameters* dialog box will appear. In Model file, browse to select the previously saved model file with extension .mdl, which is the result of training and saving teh model with a dataset. Then click OK to execute. The results of classification will be displayed in the navigator. The classification results view appears under the current spreadsheet and the results of classification are listed under it. They consist of the following views - The Classification Report, and if Class Labels are present in this dataset, the Confusion Matrix and the Lorenz Curve as well.



	CLL (Predi...	DLBCL (Pr...	FOLL (Pre...	HCL (Pred...	NLS (Predi...	Accuracy
(True) CLL	11	0	0	0	1	91.667
(True) DLBCL	0	67	1	0	0	98.529
(True) FOLL	0	0	8	0	1	88.889
(True) HCL	0	0	0	13	0	100
(True) NLS	1	1	0	1	28	90.323
Overall Accuracy						95.489

Figure 13.4: Confusion Matrix for Training with Decision Tree

13.11 Viewing Classification Results

The results of classification are shown in the four graphical views described below. These views provide an intuitive feel for the results of classification, help to understand the strengths and weaknesses of models, and can be used to tune the model for a particular problem. For example, a classification model may be required to work very accurately for one class, while allowing a greater degree of error on another class. The graphical views help tweak the model parameters to achieve this.

13.11.1 Confusion Matrix

A Confusion Matrix presents results of classification algorithms, along with the input parameters. It is common to all classification algorithms in **ArrayAssist** - algo.SVM, Neural Network, , and Decision Tree, appears as follows:

The Confusion Matrix is a table with the true class in rows and the predicted class in columns. The diagonal elements represent correctly classified experiments, and cross diagonal elements represent misclassified experiments. The table also shows the learning accuracy of the model as the percentage of correctly classified experiments in a given class divided by the total number of experiments in that class. The average accuracy of the model is also given.

- For validation, the output shows a cumulative Confusion Matrix, which

is the sum of confusion matrices for individual runs of the learning algorithm.

- For training, the output shows a Confusion Matrix of the experiments using the model that has been learnt.
- For classification, a Confusion Matrix is produced after classification with the learnt model only if class labels are present in the input data.

13.11.2 Classification Model

The classification model gives parameters related to the learning of the individual classification algorithms. Decision Trees. The model is algorithm specific and Neural Networks, SVMs, , and the details for each algorithm are given below.

Decision Tree Model

ArrayAssist implements two types of decision trees; Axis Parallel and Oblique

The Decision Tree Model shows the learnt decision tree and the corresponding table. The left panel lists the row identifiers(if marked)/row indices of the dataset. The right panel shows the collapsed view of the tree. Clicking on the Expand/Collapse Tree icon in the toolbar can expand it. The leaf nodes are marked with the Class Label and the intermediate nodes in the Axis Parallel case show the Split Attribute.

To Expand the tree Click on an internal node (marked in brown) to expand the tree below it. The tree can be expanded until all the leaf nodes (marked in green) are visible. The table on the right gives information associated with each node.

In the Axis Parallel case, the table shows the Split Value for the internal nodes. When a candidate for classification is propagated through the decision tree, its value for the particular split attribute decides its path. For values below the split attribute value, the feature goes to the left node, and for values above the split attribute, it moves to the right node. For the leaf nodes, the table shows the predicted Class Label. It also shows the distribution of features in each class at every node, in the last two columns.

For the Oblique case, the table shows the Split Equation for the internal nodes. When a candidate for classification is propagated through the decision tree, the split equation is computed with the corresponding attributes

for that node. If the value is less than zero, the experiment goes to the left node, else it moves to the right node. For the leaf nodes, the table shows the predicted Class Label. It also shows the distribution of the experiments in each class at every node.

To View Classification Click on an identifier to view the propagation of the feature through the decision tree and its predicted Class Label.



Click *Save Model* button to save the details of the algorithm and the model to an .mdl file. This can be used later to classify new data.



Expand/Collapse Tree: This is a toggle to expand or collapse the decision tree.

Neural Network Model

The Neural Network Model displays a graphical representation of the learnt model. There are two parts to the view. The left panel contains the row identifier(if marked)/row index list. The panel on the right contains a representation of the model neural network. The first layer, displayed on the left, is the input layer. It has one neuron for each feature in the dataset represented by a square. The last layer, displayed on the right, is the output layer. It has one neuron for each class in the dataset represented by a circle. The hidden layers are between the input and output layers, and the number of neurons in each hidden layer is user specified. Each layer is connected to every neuron in the previous layer by arcs. The values on the arcs are the weights for that particular linkage. Each neuron (other than those in the input layer) has a bias, represented by a vertical line into it.

To View Linkages Click on a particular neuron to highlight all its linkages in blue. The weight of each linkage is displayed on the respective linkage line. Click outside the diagram to remove highlights.

To View Classification Click on an id to view the propagation of the feature through the network and its predicted Class Label. The values adjacent to each neuron represent its activation value subjected to that particular input.

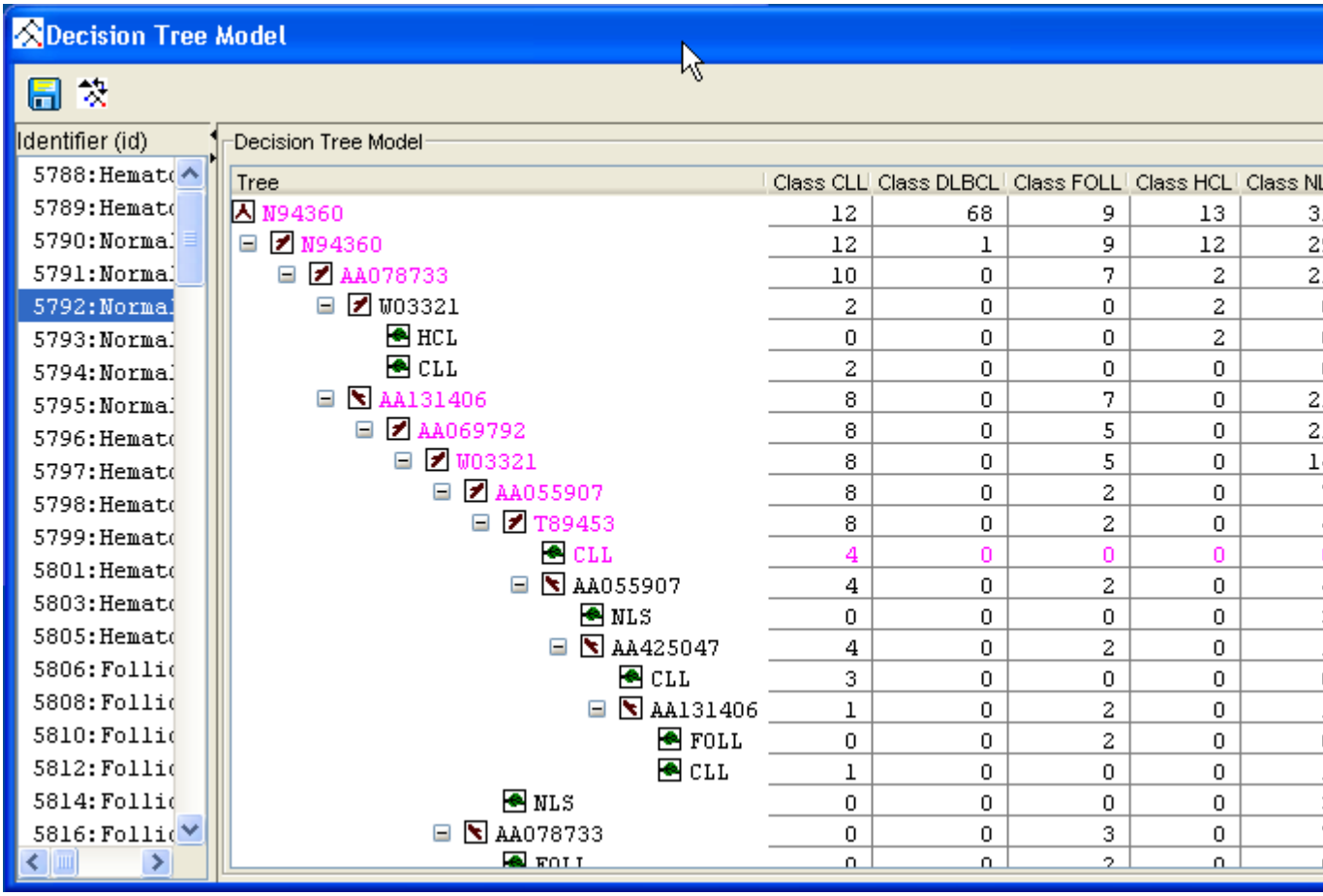


Figure 13.5: Axis Parallel Decision Tree Model

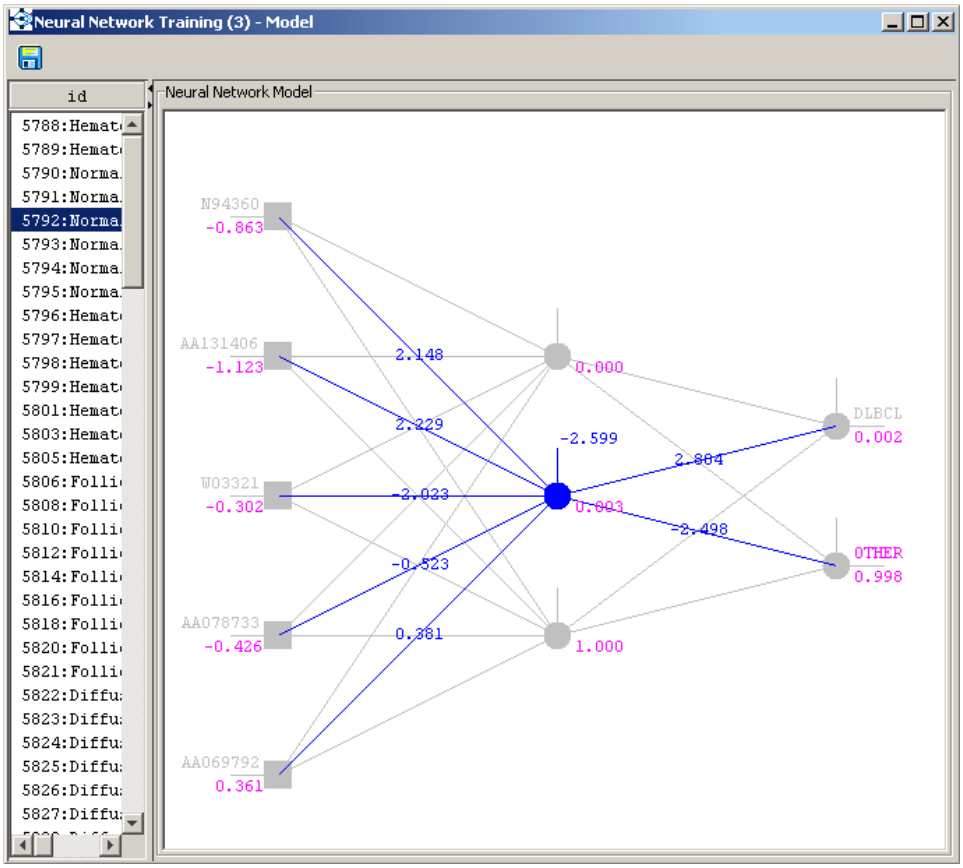


Figure 13.6: Neural Network Model



Click *Save Model* button to save the details of the algorithm and the model to an .mdl file. This can be used later to classify new data.

Support Vector Machine Model

For Support Vector Machine training, the model output contains the following training parameters in addition to the model parameters:

The top panel contains the Offset which is the distance of the separating hyperplane from the origin in addition to the input model parameters.

The lower panel contains the Support Vectors, with three columns corresponding to row identifiers(if marked)/row indices, Lagranges and Class Labels. These are input points, which determine the separating surface between two classes. For support vectors, the value of Lagrange Multipliers is non-zero and for other points it is zero. If there are too many support vectors, the SVM model has over-fit the data and may not be generalizable.



Click *Save Model* button to save the model to a .mdl file. This can be used later to classify new data.

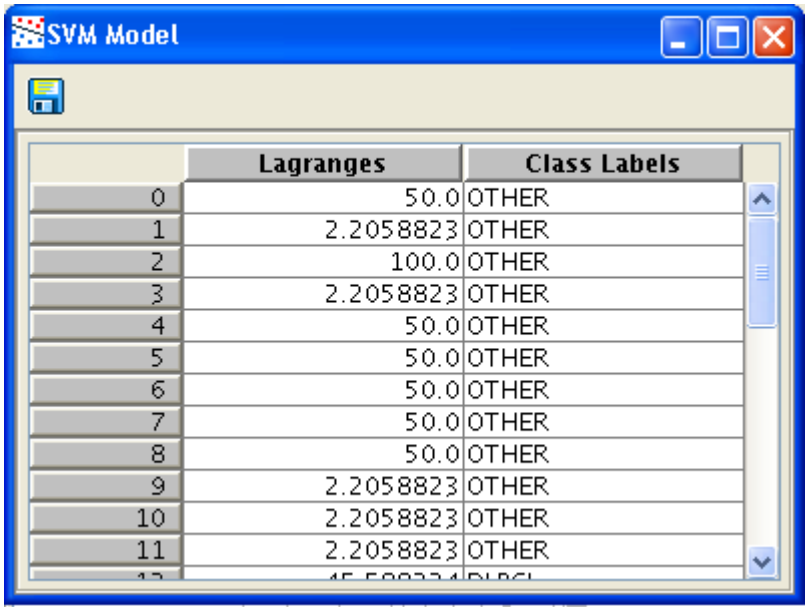
13.11.3 Classification Report

This report presents the results of classification. It is common to the three classification algorithms - Support Vector Machine, Neural Network, and Decision Tree.

The report table gives the identifiers; the true Class Labels (if they exist), the predicted Class Labels and class belongingness measure. The class belongingness measure represents the strength of the prediction of belonging to the particular class.

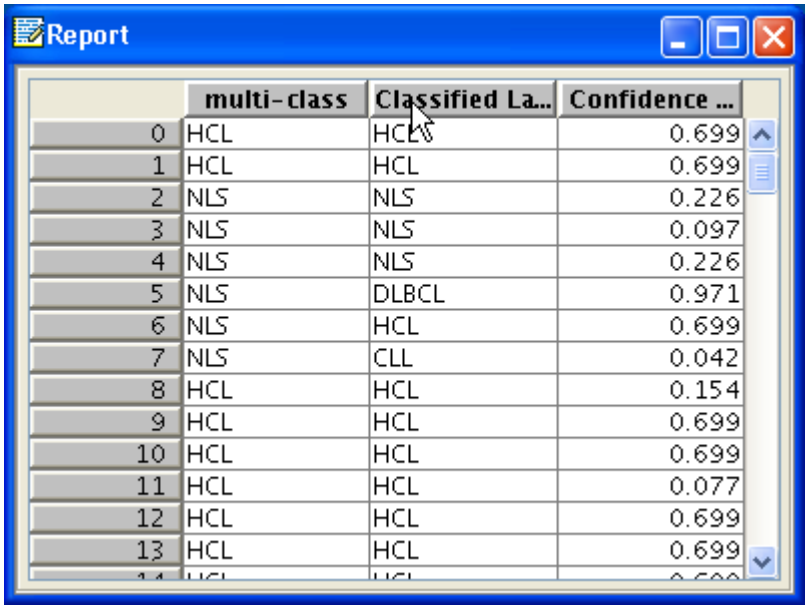
Report Operations

Save Report to File Right click anywhere in the report windows and choose *Export As* →*Text* option to save the report to a tab-delimited ASCII text file.



	Lagranges	Class Labels
0	50.0	OTHER
1	2.2058823	OTHER
2	100.0	OTHER
3	2.2058823	OTHER
4	50.0	OTHER
5	50.0	OTHER
6	50.0	OTHER
7	50.0	OTHER
8	50.0	OTHER
9	2.2058823	OTHER
10	2.2058823	OTHER
11	2.2058823	OTHER

Figure 13.7: Model Parameters for Support Vector Machines



	multi-class	Classified La...	Confidence ...
0	HCL	HCL	0.699
1	HCL	HCL	0.699
2	NLS	NLS	0.226
3	NLS	NLS	0.097
4	NLS	NLS	0.226
5	NLS	DLBCL	0.971
6	NLS	HCL	0.699
7	NLS	CLL	0.042
8	HCL	HCL	0.154
9	HCL	HCL	0.699
10	HCL	HCL	0.699
11	HCL	HCL	0.077
12	HCL	HCL	0.699
13	HCL	HCL	0.699

Figure 13.8: Decision Tree Classification Report

Export Columns to Dataset The *Predicted Class* and *Class belongingness* columns can be exported back to the dataset to be used in other views and subsequent algorithms and commands. Select a column by Left-Click anywhere inside it. The column is highlighted in the selection color. Click on the *Export Column* button in the top-level toolbar (or Right-Click and choose *Export Column* menu) to append this column back to the dataset. An information message appears when a column is successfully appended to the dataset in this manner. NOTE: The first two columns cannot be exported to the dataset since they do not reveal any additional information and are already a part of the dataset columns.

13.11.4 Lorenz Curve

Predictive classification in **ArrayAssist** is accompanied by a class belongingness measure, which ranges from 0 to 1. The Lorenz Curve is used to visualize the ordering of this measure for a particular class.

The items are ordered with the predicted class being sorted from 1 to 0 and the other classes being sorted from 0 to 1 for each class. The Lorenz Curve plots the fraction of items of a particular class encountered (Y-axis) against the total item count (X-axis). The blue line in the figure is the ideal curve and the deviation of the red curve from this indicates the goodness of the ordering.

For a given class, the following intercepts on the X-axis have particular significance:

The light blue vertical line indicates the actual number of items of the selected class in the dataset.

The light red vertical line indicates the number of items predicted to belong to the selected class.

Classification Quality The point where the red curve reaches its maximum value (Y=1) indicates the number of items which would be predicted to be in a particular selected class if all the items actually belonging to this class need to be classified correctly.

Consider a dataset with two classes A and B. All points are sorted in decreasing order of their belongingness to A. The fraction of items classified as A is plotted against the number of items, as all points in the sort are traversed. The deviation of the curve from the ideal indicates the quality

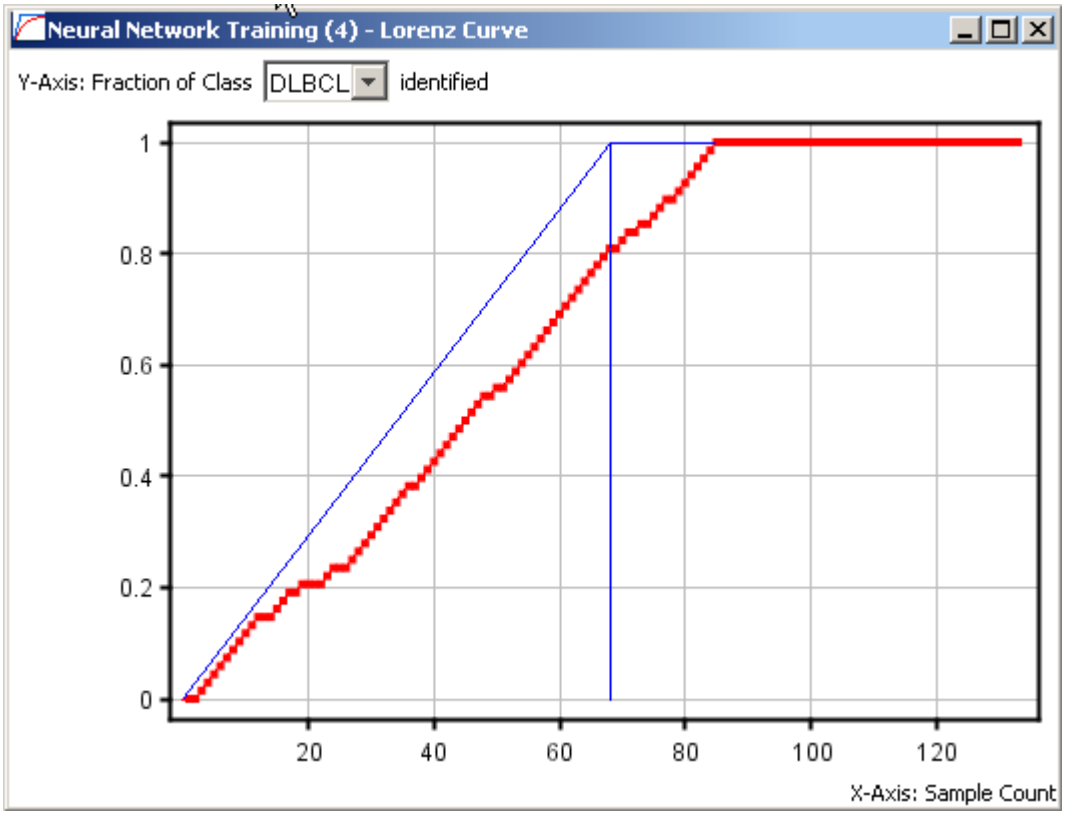


Figure 13.9: Lorenz Curve for Neural Network Training

of classification. An ideal classifier would get all points in A first (linear slope to 1) followed by all items in B (flat thereafter). The Lorenz Curve thus provides further insight into the classification results produced by **ArrayAssist**. The main advantage of this curve is that in situations where the overall classification accuracy is not very high, one may still be able to correctly classify a certain fraction of the items in a class with very few false positives; the Lorenz Curve allows visual identification of this fraction (essentially the point where the red line starts departing substantially from the blue line).

Lorenz Curve Operations

The Lorenz Curve view is a lassoed view and is synchronized with all other lassoed views open in the desktop. It supports all selection and zoom operations like the scatter plot.

Class Selection Use the Y-Axis dropdown combobox to choose the class for which the Lorenz Curve is displayed.

13.12 Guidelines for Classification Operations

Classification algorithms are complex and need considerable experimentation and experience to fully exploit their power. To train a model, it is essential to have a column marked as a Class Label column in the spreadsheet. It is important to visualize and explore the data before using classification algorithms. If the classes look clustered and clearly separable in the scatter plots and PCA plots, then there is a good chance that a classification model would be effective in classifying the data. In general, it is better to use a simple model for learning from the data to avoid over-fitting. Thus the linear kernel SVM or the axes parallel decision tree would be the first algorithms to try.

For two class data, any of the algorithms can be used, while for multi-class data, only Neural Networks or Axis Parallel Decision Trees can be used. Only Decision Trees allow the use of Categorical variables (string columns and integer columns explicitly marked as categorical; default for integers is continuous). Finally, if continuous values and not discrete classes need to be learnt then use the Linear regression algorithm.

13.13 Table of Advantages, Disadvantages of Classification Algorithms

13.14 What is the Recommended Sequence of using Algorithms

This is a difficult question. Generally, classification is an interactive process in which the user has to make decisions at many points. Overall, a normal sequence would be to run Validation with all the algorithms and tweak various parameters. Once you are satisfied with the Confusion Matrix and

Algorithm	No. Classes	Speed	Memory	Model Inference	Convergence
Axis Parallel Decision Tree	≥ 2	Fast	Low	Intuitive	Irrelevant
Oblique Decision Tree	2	Slow	Low	Intuitive	Data Dependent
Support Vector Machine	2	Medium	High	Mathematical	Data Dependent
Neural Network	≥ 2	Slow	Medium	Graphical	Data Dependent
Naive Bayesian Classifier	≥ 2	Medium	Medium	Graphical	Irrelevant

Table 13.2: Table of Performance of Classification Algorithms

errors, run Train with the best parameters. This would yield a model that should be saved to be re-used for classifying new data.

In general, the algorithms can be tried with the following sequence. First, try Axis Parallel Decision Trees, SVM with a linear kernel and neural network with zero hidden layers. These are simple linear classifiers and may work in most cases. If these are not satisfactory, try the Oblique Decision Trees, SVM with Polynomial and Gaussian kernels, and Neural Network with more than one hidden layer (say, three hidden layers with 7,5,3 neurons respectively, works well in several cases).

13.15 Typical Cases Explained with Various Views

Example: Iris Dataset Iris is a time-honored dataset used by Fisher as an example for discriminant analysis. Since then, it has been used extensively for clustering and classification problems, as well as being included in many learning dataset repositories. This dataset is included here for testing the analysis tools in **ArrayAssist**. It is a small dataset with 150 rows and 4 columns containing measurements of sepal width, sepal length, petal width and petal length of three sub-species of Iris flowers.

- Load the iris.csv dataset from the samples directory and mark the flower column (first column) as the Class Label column.
- View the data for classification in the matrix plot. This shows a clear separation of Iris-setosa from Iris-versicolor and Iris-virginica. This

separation is clearer in the petal length and petal width dimensions. Any linear classifier should be able to learn separation of the two classes. Try the SVM with linear kernel after converting the classification problem to a binary classification problem. Neural Network with no hidden layers can also be used.

- Neural Network seems to separate the data into two classes, while the third class, versicolor, appears to get distributed between these two classes. The separation between versicolor and virginica is not very clear and the two are intermixed. The plots show that versicolor and viginica may be separable in axis parallel cuts of the data.
- Try the Axis Parallel Decision Tree and examine the results. Expand the Decision Tree model. It is clear that only petal width and petal length have been used to obtain an accuracy of over 97% with only three misclassifications. Examine the Lorenz Curve. The misclassifications are near the boundaries of the classifier, which is shown in the scatter plot.
- It will be interesting to try validation with different options to examine how generalizable the classification model is. However, since the sample size is small, it might be judicious to use leave-one-out or 5-fold validation methods, so that there are an adequate number of samples for training. Examine the results of validation, train with the same set of parameters, and save the model for classifying a new flower, based on flower size measurements.

Example: Lymphoma Dataset http://llmpp.nih.gov/lymphoma/ Alizadeh et al, Nature 403, 2000)included in the samples directory contains expressions of 13,999 genes from experimental samples of different types of lymphoma. The intent of the experiment is to identify genes that are expressed in different types of lymphoma, and to predict and differentiate between Diffuse Large B-Cell Lymphoma (DLBCL) and all other types.

Use lymph1000.csv. The classification problem is to differentiate the gene expression profiles of DLBCL from the rest. The data is preprocessed and filtered for missing values (which are filled in with the value 0) and for low variation. Then it is transposed so that the rows contain samples and columns contain genes. Two Class Label columns are present in this dataset (the last column and the second last column; they are called two-class and multi-class respectively). The pre-processed and transposed data, with only 1000 feature selected genes, is stored in lymph1000.csv. The following exercise explores this dataset.

- Load lymph1000.csv from the samples directory and mark the last column as the Class Label column. The dataset has experiments as rows and 1000 genes as columns.

- View the data for classification in a PCA plot. It shows a possible separation of data even when only 34.7% of the variation is captured by the first two principal axes. The Eigen Values curve descends sharply, suggesting that six or seven principal axes would capture all the variation. It might be interesting to try transforming the data into Eigen space and running classification.

- Validate with SVM, Neural Network and Decision Tree with their default parameters. Only the results of the Axis Parallel Decision Tree look promising. This might be due to the presence of redundant data. This is why the Neural Network is slow and SVM does not yield good results.

- Use feature selection to remove redundant genes that do not discriminate between DLBCL and other lymphoma cells. Run the Kruskal-Wallis test and save the top 10 features that have the smallest p-value. A dataset with these 10 features has been saved into lymph10.csv. Run validation with all three algorithms using default parameters with the new data. Compare the confusion matrices of the two runs. The Neural Network is much quicker, since there are a smaller number of columns in the data now, and yields better results. Axis Parallel Decision Tree yields a similar result.

- Run train using default parameters of all three algorithms on the 20-feature dataset. Examine the confusion matrices. The results of all three algorithms are satisfactory. Examine the Axis Parallel Decision Tree model and expand the tree. The learnt tree is small with only two genes - N94360 and AA131406. These are the two important genes that differentiate between DLBCL and other lymphomas. Axis Parallel Decision Tree can, therefore, also be used for identifying features to be used by other training algorithms.

- Run Axis Parallel Decision Tree for the multi class problem of classifying all types of lymphomas in the dataset lymphoma1000.csv. Class Labels have already been created. Mark the Class Label column in the spreadsheet and run the Axis Parallel Decision Tree. Examine the results.

Reference

Jeannette Lawrence. Introduction to Neural Networks. California Scientific Software. 1993

N. Christianini and J. Shawe Taylor. An Introduction to Support Vector Machines. Cambridge University Press.

Chapter 14

Regression: Learning and Predicting Outcomes

14.1 What is Regression

The *Classification* chapter discussed training and prediction of models for classifying input into discrete classes. This chapter describes the technique of *Regression* which is used when the Class Labels are continuous-valued instead of discrete-valued. Thus, to predict whether a tumor sample is cancerous or not, one would use one of the previous four classification methods, but to predict the survival index value associated with a particular sample, one would use the regression method. This method treats the Class Label column as a continuous variable and tries to find a a function in the feature space which predicts the label with least error.

Model building for regression in **ArrayAssist** is done using two powerful algorithms - Multivariate Linear Regression (MLR). Neural Network (NN), Models built with these algorithms can then be used to predict continuous values.

14.2 Regression Pipeline Overview

14.2.1 Dataset Orientation:

All classification and prediction algorithms in **ArrayAssist** predict classes/values for rows in the dataset. Therefore, when predicting gene function classes, genes should be along rows and samples/experiments along columns. And when predicting phenotypic properties of samples based on gene expression,

samples should be along rows and genes should be along columns. To get the right orientation, use the Transpose feature available from Data → Transpose if necessary. This will create a new dataset in a new datatab that can be using for classification.

14.2.2 Class Labels and Training:

The next step, to learn a model from the data in the spreadsheet, Training needs to be performed using one of the algorithms available. For training, each row needs to have an associated Class Label which describes the value of the phenotypic variable associated with the row. These values must appear in a special column which contains the Class Labels. This column can be specified before execution by specifying the appropriate column in the *Columns* section of Algorithm Parameters dialog. This is a frequently needed operation, so a convenient way is provided to permanently mark a column as a Class Label column in the dataset. See the Creating a Class Label column heading below to see how existing columns can be marked as Class Label columns, or how a new Class Label column is created.

Once the Class Label column is set up, training can be run using one of the several learning algorithms available in **ArrayAssist**. This process will mine the data and come up with a model which can be saved in a file for future use. The actual meaning and representation of this model varies with the method used. The training process also comes up with a variable value for each of the rows as predicted by the model being constructed. These predictions give some feel for how good the model is. However, it is dangerous to trust models based on these predictions as the training process often has a tendency to over-fit, i.e. yield models which memorize the data. If this is indeed the case then these models will not work well when predicting on new data with unknown Class Labels.

14.2.3 Feature Selection:

Very often, model prediction accuracies and algorithm speeds can be substantially increased by performing training not with the whole feature set but with only a subset of relevant and important features. Several tests for selecting important features are available in **ArrayAssist**. Once the dataset is restricted to these features, this feature set needs to be validated, as above.

Features and Validation:

To give a feel for how well a model obtained in the training step would do in the prediction step on a new dataset, we need to run Validate on the feature set. The feature set is the set of columns in the dataset. The aim in validation is to check whether the given set of features in the dataset is powerful enough to yield good models which can make accurate predictions on new datasets. In the absence of this new dataset, the existing dataset is split into two parts by the validation process - one part is used for training; the resulting model is applied on the second part, and the errors of the predictions are output. If these predictions have less errors, then the feature set is a good one and the model obtained in training is likely to perform well on new datasets, provided of course that the training dataset captures the distributional variations in these new datasets.

14.2.4 Regression:

If the validation error obtained above is low then training can be used to build a model which will then be used for prediction on new datasets. High validation accuracies indicate that this model is likely to work well in practice.

14.3 Specifying a Class Label Column

Training and validation require that all rows have Class Labels associated with them. The column containing the Class Labels can be specified before execution by specifying the appropriate column in the *Columns* section of Algorithm Parameters dialog. This is a frequently needed operation, and the Class Label column is used in several other visualizations as well; so a convenient way is provided to permanently mark a column as a Class Label column in the dataset.

Specifying a Class Label Column in the dataset An existing column can be permanently marked as the Class Label column in the dataset using the *Mark* command. Click the *Mark* icon in the spreadsheet toolbar (or select Data →Mark option) and specify an existing column as Class Label column.

Creating a new Class Label Column If a Class Label column does not already exist in the dataset, then there are multiple ways to create a new Class Label column.

- Use the *Create New Column Using Formula* command to append a new column to the dataset with the appropriate values. This command is accessible from the *Create New Column* icon in the spreadsheet toolbar, as well as Data ⟶Column Operations⟶Create New Column menu item.
- Import the columns from a file and mark them as class label.

14.4 Selecting features for Regression

Very often, model prediction accuracies and algorithm speeds can be substantially increased by performing training not with the whole feature set but with only a subset of relevant and important features. Several tests for selecting important features are available in **ArrayAssist**. Once the dataset is restricted to these features, this feature set needs to be validated, as above. In addition, feature selection becomes necessary when the number of features (columns) exceeds the number of samples (rows). In such cases, the differentiating features must be separated out from the non-differentiating features, and these should be the only ones used from training and prediction.

ArrayAssist supports two statistical tests to help select important features for regression and reduce the dimensionality of the data. These tests are done on all features (i.e. columns of data). They check which features are highly correlated and produce an associated significance or p-value for each feature, ranked in decreasing order of correlation. The basic premise is that it should be able to pick up only one feature from a set of highly correlated features, and that feature represents this set in the training and classification process.

14.4.1 Correlation

This test computes a *Pearson Correlation Coefficient* for every selected column with respect to a user-specified reference column, and ranks all columns in decreasing order of absolute value of correlation, assuming that the values in each column are normally distributed. Visualizing the distribution of all columns using Descriptive Statistics will give a rough indication whether columns values are normally distributed. If the distribution is not normal, the non-parametric *Rank Correlation* test may be more appropriate.

To select features using Correlation:

- Select *Regression* → *Feature Selection* → *Correlation* option. Choose the input set of columns from the *Columns* tab in the dialog, and spec-

ify a reference column in the parameters tab. Click OK to execute the command. The results appear in a window titled *Correlation Feature Ranking*. The results consists of three columns. The first column contains column names sorted in decreasing order of correlation. The second column gives the respective *Pearson Correlation Coefficient* value ($\mathcal{R}^2$), and the third column gives the p-value.

Based on this analysis, features can be selected and saved to a file, or a new dataset can be created for further classification analysis. Features can be selected based on the p-value, or the rank of the p-value, as explained in Saving Features and Creating New Datasets section.

14.4.2 Rank Correlation

This test computes a *Spearman Correlation Coefficient* for every selected column with respect to a user-specified reference column, and ranks all columns in decreasing order of correlation. It is essentially similar to the *Correlation* method, but uses the ranks instead of actual values. This eliminates the assumption of normally distributed values

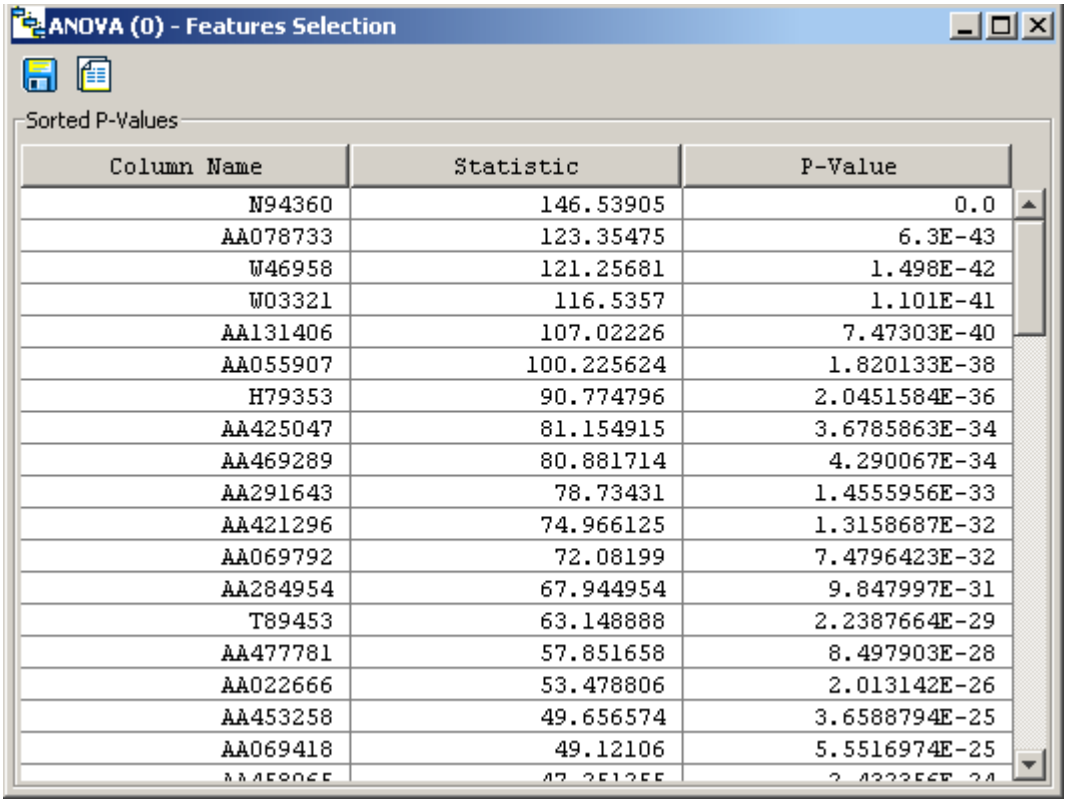
To select features using Rank Correlation:

- Select *Regression* → *Feature Selection* → *Correlation* option. Choose the input set of columns from the *Columns* tab in the dialog, and specify a reference column in the parameters tab. Click OK to execute the command. The results appear in a window titled *Correlation Feature Ranking*. The results consists of three columns. The first column contains column names sorted in decreasing order of correlation. The second column gives the respective *Spearman Correlation Coefficient* value ($\mathcal{R}^2$), and the third column gives the p-value.

  Based on this analysis, features can be selected and saved to a file, or a new dataset can be created for further classification analysis. Features can be selected based on the p-value, or the rank of the p-value, as explained in Saving Features and Creating New Datasets section.

14.5 The Three Steps in Regression

Building a regression model involves experimenting with different algorithms and parameters. Regression in **ArrayAssist** has three components - Train, Validate and Predict. Training involves using a dataset with known class



Column Name	Statistic	P-Value
N94360	146.53905	0.0
AA078733	123.35475	6.3E-43
W46958	121.25681	1.498E-42
W03321	116.5357	1.101E-41
AA131406	107.02226	7.47303E-40
AA055907	100.225624	1.820133E-38
H79353	90.774796	2.0451584E-36
AA425047	81.154915	3.6785863E-34
AA469289	80.881714	4.290067E-34
AA291643	78.73431	1.4555956E-33
AA421296	74.966125	1.3158687E-32
AA069792	72.08199	7.4796423E-32
AA284954	67.944954	9.847997E-31
T89453	63.148888	2.2387664E-29
AA477781	57.851658	8.497903E-28
AA022666	53.478806	2.013142E-26
AA453258	49.656574	3.6588794E-25
AA069418	49.12106	5.5516974E-25

Figure 14.1: Feature Selection Output

values, and learning a model from that dataset. However, models that fit the training dataset very well may fail for new data points. Such over-fitting of the training data will most likely yield a model that cannot be generalized and, therefore, would not be useful. Therefore, an algorithm and its associated parameters must be validated before they are used to predict new data. This process involves segmenting the training data into two sets. One set is used for training and the other for testing the model. Typically, validation should be done with a variety of algorithms and model parameters, and results monitored to choose the best combination. This combination can then be used to build a model with the entire training dataset, and then to predict new data.

14.5.1 Validate

Validation helps to choose the right set of features, an appropriate algorithm and associated parameters for a particular dataset. Validation is also an important tool to avoid over-fitting models on training data as over-fitting will give low accuracy on validation. Validation can be run on the same dataset using various algorithms and altering the parameters of each algorithm. The results of validation, presented in a report, are examined to choose the best algorithm and parameters for the regression model.

N-fold: The rows in the input data are randomly divided into N equal parts; N-1 parts are used for training, and the remaining one part is used for testing. The process repeats N times, with a different part being used for testing in every iteration. Thus each row is used at least once in training and once in testing, and a prediction for every row is obtained. This whole process can then be repeated as many times as specified by the number of repeats. Mean and Standard deviation of predictions for a row in different repeats is reported in the validation report. Mean values of the predictions are used to compute Mean-Absolute-Error, Maximum-Absolute-Error, Root-Mean-Squared-Error and Q2 for validation. These statistics are also reported statistical results.

The default values of three-fold validation and ten repeat should suffice for most approximate analysis. Higher number of repeats give a stable estimate of mean and standard deviation for the predictions.

14.5.2 Train

Each of the learning algorithms in **ArrayAssist** can be trained with a (hopefully representative) dataset that has Class Labels. The results of training yield a Model, a Report, a Statistical Report

14.5.3 Prediction

Prediction applies the regression model to a new dataset and generates a new column of predicted values and the associated confidence in prediction. To run Prediction, select *Regression* → *Predict* menu and specify a model file generated from the training step. Click on OK to begin execution.

The output of the prediction process is a *Prediction Report* which displays the predicted value of the dependent variable in a tabular format. This report also has a confidence for prediction in case of linear regression. Both of these columns can then be exported back to the dataset by clicking on the *Export Column* button in the main toolbar or by accessing the same through the Right-Click popup menu. The report can also be saved in a tabular form to a tab-separated ASCII text file using the *Export* → *Text* option in the Right-Click menu.

14.6 Multivariate Linear Regression

Multivariate Linear Regression fits a function that uses linear combination of features to predict the label with least sum of squares error. Linear Regression over-fits when the number of features is greater than the number of rows and is therefore allowed only on datasets where the number of columns (features) is at most the number of rows.

14.6.1 Linear Regression Train

Once the desired set of *good* features and samples is ready, a model is trained to predict a continuous value as a linear combination of features. Linear multivariate regression model is represented by

$$y = \Sigma \alpha_i x_i + c$$

where y is the dependent variable being regressed, and $x_0, x_1, ...$ are the features, and $\alpha_0, \alpha_1, ...$ are the weights associated with the features.

Select Regresssion → Train menu to invoke training. The following options are available for training.

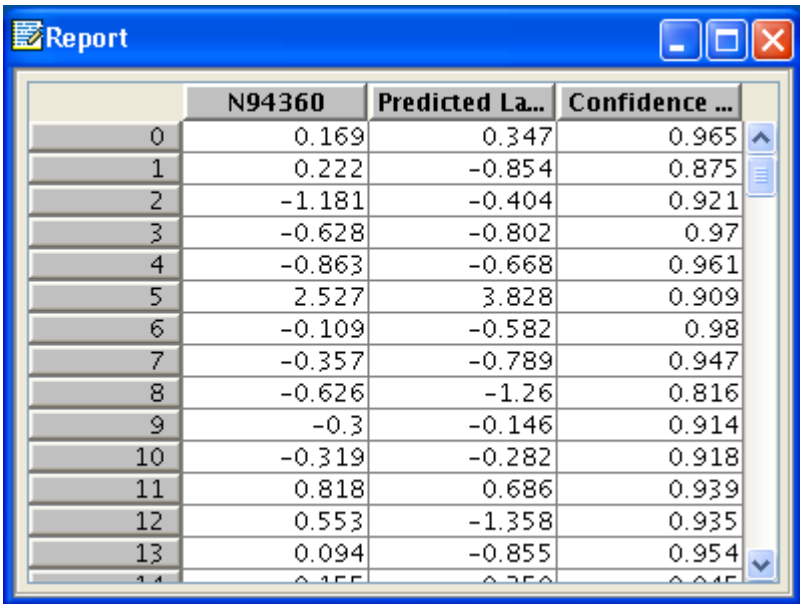


Figure 14.2: Linear Regression Training Report

Regressed Column Specify the Class Label column (i.e. the dependent variable) in the drop down combo box.

Fit line without intercept Constrains regression equation with $c = 0$ (i.e. the constant must be zero).

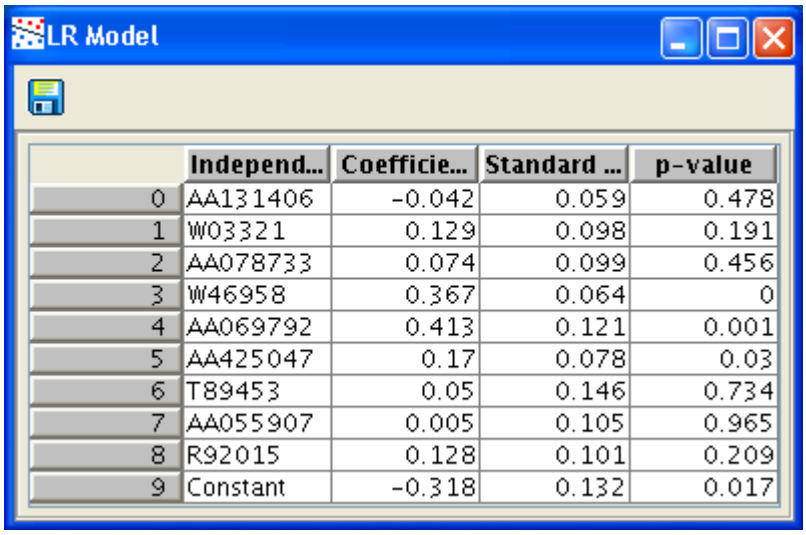
The training algorithm essentially determines the weights (and the constant) such that the RMS error for the predicted value is the least possible. The output consists of a model, report and an error model.

- Linear Regression Report

  The report table gives the identifiers; the true value, the predicted value from the regression equation and confidence in each prediction. The report can either be saved to an ASCII text file, or the *Predicted Value* and *Residual* columns can be exported back to the dataset as described in section Report Operations.

- Linear Regression Model

  The model consists of the weights $\alpha_0, \alpha_1, ...$ for every feature as well as the constant value. Click on the *Save Model* button to save this model



	Independ...	Coefficie...	Standard ...	p-value
0	AA131406	-0.042	0.059	0.478
1	W03321	0.129	0.098	0.191
2	AA078733	0.074	0.099	0.456
3	W46958	0.367	0.064	0
4	AA069792	0.413	0.121	0.001
5	AA425047	0.17	0.078	0.03
6	T89453	0.05	0.146	0.734
7	AA055907	0.005	0.105	0.965
8	R92015	0.128	0.101	0.209
9	Constant	-0.318	0.132	0.017

Figure 14.3: Linear Regression Model

to a file for use in prediction later. The model can also be exported to a tab-separated ASCII text file by selecting the *Export* → *Text* option in the Right-Click popup menu.

- Statistical Error Model:

  The error model provides useful information about the accuracy of the fit achieved by the model. It provides several standard statistical error-estimates which help in pinning down the accuracy of the generated regression model. The error model can also be exported to a human readable ASCII text file by selecting the *Export* → *Text* option in the Right-Click popup menu.

- The Analysis of Variance (ANOVA) Table:

  The ANOVA table partitions the variance in the response variable into two parts. One portion is accounted by the model. The remaining portion is the variance that remains even after the model is used. The model is considered to be statistically significant if it can account for a large amount of variance in the response.

  The column labelled Source in ANOVA table has three rows. One for total variance and one for each of the two pieces that is, Regression and Error.

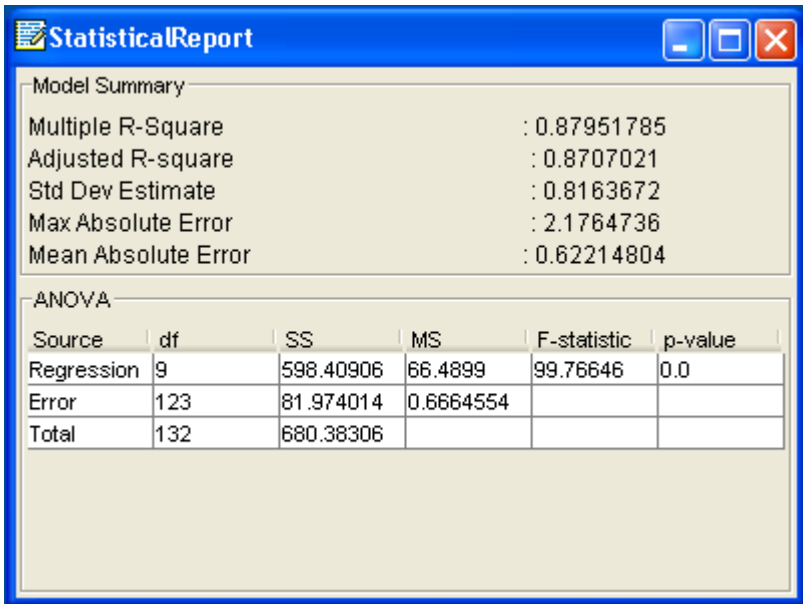


Figure 14.4: Linear Regression Error Model

Sums of Squares: The total amount of variance in the response can be written

$$\sum_i (y_i - \bar{y})^2$$

where $\bar{y}$ is the sample mean. When the regression model is used for prediction, the amount of uncertainty that remains is the variance about the regression line, $\sum_i (y_i - \hat{y}_i)^2$ where $\hat{y}_i$ is the predicted i^{th} response. This is the Error sum of squares. The difference between the Total sum of squares and the Error sum of squares is the Model Sum of Squares, which happens to be equal to

$$\sum_i (\hat{y}_i - \bar{y})^2$$

Each sum of squares has corresponding degrees of freedom (DF) associated with it. Total df is one less than the number of observations, $n - 1$. The Model df is the number of independent variables in the

model, p. The Error df is the difference between the Total df $n-1$ and the Model df p, that is, $n-p-1$.

The Mean Squares are the Sums of Squares divided by the corresponding degrees of freedom.

The F Value or F ratio is the test statistic used to decide whether the model as a whole has statistically significant predictive capability, considering the number of variables needed to achieve it. F is the ratio of the Model Mean Square to the Error Mean Square. Under the null hypothesis that the model has no predictive capability, the F statistic follows an F distribution with p numerator degrees of freedom and $n-p-1$ denominator degrees of freedom. The null hypothesis is rejected if the F ratio is large.

The F-test associated with the ANOVA table tests

$$H_0 : \alpha_0 = \alpha_1 = \alpha_m = 0$$

against

$$H_A : \alpha_i \neq 0 for i = 0, 1...m$$

Null Hypothesis says that there is no linear relationship between the mean of y and any subset of the explanatory variables x_i

R^2 is the squared multiple correlation coefficient. It is also called the Coefficient of Determination. R^2 is the ratio of the Regression sum of squares to the Total sum of squares, $RegSS/TotSS$. It is the proportion of the variability in the response that is accounted for by the model. Since the Total SS is the sum of the Regression and Residual Sums of squares, R^2 can be rewritten as

$$(TotSS - ResSS)/TotSS = 1 - ResSS/TotSS$$

Some call R^2 the proportion of the variance explained by the model. If a model has perfect predictability, $R^2 = 1$. If a model has no predictive capability, $R^2 = 0$.

As additional variables are added to a regression equation, R^2 increases even when the new variables have no real predictive capability. The adjusted-R^2 is an R^2 like measure that avoids this difficulty. When

variables are added to the equation, adj-R^2 doesn't increase unless the new variables have additional predictive capability.

$$adjR^2 = 1 - (ResSS/ResDF)/(TotSS/(n-1))$$

Additional variables with no explanatory capability may increase the Regression SS (and reduce the Residual SS) but they will not decrease the standard error of the estimate. The reduction in Residual SS will be accompanied by a decrease in Residual DF. If the additional variable has no predictive capability, these two reductions will cancel each other out.

The Root Mean Square Error(RMSE) is the square root of the Residual Mean Square. It is the standard deviation of the data about the regression line, rather than about the sample mean.

The Standard Errors are the standard errors of the regression coefficients. They can be used for hypothesis testing and constructing confidence intervals. The degrees of freedom used to calculate the P values is given by the Error DF from the ANOVA table. The P values tell us whether a variable has statistically significant predictive capability in the presence of the other variables, that is, whether it adds something to the equation. In some circumstances, a non-significant P value might be used to determine whether to remove a variable from a model without significantly reducing the model's predictive capability. For example, if one variable has a non-significant P value, we can say that it does not have predictive capability in the presence of the others,remove it, and refit the model without it. These P values should not be used to eliminate more than one variable at a time, however. A variable that does not have predictive capability in the presence of the other predictors may have predictive capability when some of those predictors are removed from the model.

NOTE: Training will fail to produce a model in two cases

- When the number of features is greater than number of samples; i.e the number of selected columns is greater than the number of rows. Use feature selection to reduce feature count in this case.
- When the features have a strong linear dependency between each other. This produces a singularity in the solution, and regression will fail with an error message. Remove a few strongly inter-dependent features and try running training again in this case.

14.6.2 Linear Regression Validate

To validate, select Linear Regression from the Regression drop down menu and choose Validate. The *Parameters* dialog box for Linear Regression Validation will appear. In addition to the parameters explained above for Linear Regression training, the following validation specific parameters need to be specified.

Number of Folds If N-Fold is chosen, specify the number of folds. The default is 3.

Number of Repeats The default is 1.

The results of validation with Linear Regression are displayed in the navigator. The Linear Regression view appears under the current spreadsheet and the results of validation are listed under it. They consist of the a Report and Statistical report described below:

- Regression Report

  The report table gives the identifiers; the true value, the mean and standard deviation of predicted values across all repeats. The report can either be saved to an ASCII text file, or the *Predicted Value* and *Residual* columns can be exported back to the dataset.

- Statistical Report This report gives the mean absolute error, maximum absolute error and Root-Mean-Squared error for mean predicted values. It also report R2 computed on the mean predicted values.

14.7 Neural Network

Neural Networks can handle non-linearity in relationships between features and class-labels. The Neural Network implementation in **ArrayAssist** is the multi-layer perceptron trained using the back-propagation algorithm. It consists of layers of neurons. The first is called the input layer and features for a row to be classified are fed into this layer. The last is the output layer which has an output node for the predicted value. Each neuron in an intermediate layer is interconnected with all the neurons in the adjacent layers.

The strength of the interconnections between adjacent layers is given by a set of weights which are continuously modified during the training stage using an iterative process. The rate of modification is determined by a

constant called the learning rate. The certainty of convergence improves as the learning rate becomes smaller. However, the time taken for convergence typically increases when this happens. The momentum rate determines the effect of weight modification due to the previous iteration on the weight modification in the current iteration. It can be used to help avoid local minima to some extent. However, very large momentum rates can also push the neural network away from convergence.

The performance of the neural network also depends to a large extent on the number of hidden layers (the layers in between the input and output layers) and the number of neurons in the hidden layers. Neural networks which use linear functions do not need any hidden layers. Nonlinear functions need at least one hidden layer. There is no clear rule to determine the number of hidden layers or the number of neurons in each hidden layer. Having too many hidden layers may affect the rate of convergence adversely. Too many neurons in the hidden layer may lead to over-fitting, while with too few neurons the network may not learn.

The following sections give Neural Network parameters for training, validation and classification.

14.7.1 Neural Network Train

To train a Neural Network, select the Neural Network algorithm from the Regression menu and choose Train. The *Parameters* dialog box for Neural Network will appear. The training input parameters to be specified are as follows:

Number of Layers Specify the number of hidden layers, from layer 0 to layer 9. The default is layer 0, i.e., no hidden layers. In this case, the Neural Network behaves like a linear classifier.

Set Neurons This specifies the number of neurons in each layer. The default is 3 neurons. Vary this parameter along with the number of layers.

Starting with the default, increase the number of hidden layers and the number of neurons in each layer. This would yield better training accuracies, but the validation accuracy may start falling after an initial increase. Choose an optimal number of layers, which yield the best validation accuracy. Normally, up to 3 hidden layers are sufficient. A typical configuration would be 3 hidden layers with 7,5,3 neurons, respectively.

Number of Iterations The default is 100 iterations. This is normally adequate for convergence.

Learning Rate The default is a learning rate of 0.7. Decreasing this would improve chances of convergence but increase time for convergence.

Momentum The default is a 0.3.

The results of training with Neural Network are displayed in the navigator. The Neural Network view appears under the current spreadsheet and the results of training are listed under it. They consist of the Neural Network model with parameters which can be saved as an .mdl file, a Report, a Statistical Report

- Neural Network Model

  The Neural Network Model displays a graphical representation of the learnt model. There are two parts to the view. The left panel contains the row identifier(if marked)/row index list. The panel on the right contains a representation of the model neural network. The first layer, displayed on the left, is the input layer. It has one neuron for each feature in the dataset represented by a square. The last layer, displayed on the right, is the output layer. It has one neuron for each class in the dataset represented by a circle. The hidden layers are between the input and output layers, and the number of neurons in each hidden layer is user specified. Each layer is connected to every neuron in the previous layer by arcs. The values on the arcs are the weights for that particular linkage. Each neuron (other than those in the input layer) has a bias, represented by a vertical line into it.

  To View Linkages Click on a particular neuron to highlight all its linkages in blue. The weight of each linkage is displayed on the respective linkage line. Click outside the diagram to remove highlights.

  To View Prediction Click on an id to view the propagation of the feature through the network and its predicted Class Label. The values adjacent to each neuron represent its activation value subjected to that particular input.



Click *Save Model* button to save the details of the algorithm and the model to an .mdl file. This can be used later to predict on new data.

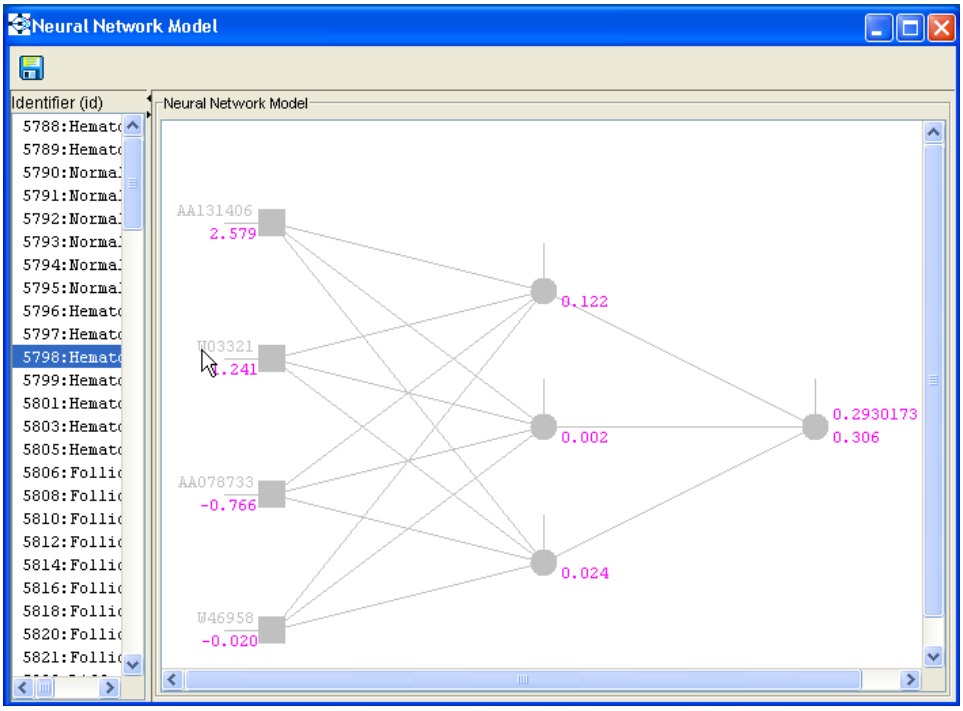


Figure 14.5: Neural Network Model

- Regression Report

  The report table gives the identifiers; the true value, the mean and standard deviation of predicted values across all repeats. The report can either be saved to an ASCII text file, or the *Predicted Value* and *Residual* columns can be exported back to the dataset as described in section Report Operations.

- Statistical Report This report gives the mean absolute error, maximum absolute error and Root-Mean-Squared error for mean predicted values. It also report R2 computed on the mean predicted values.

14.7.2 Neural Network Validate

To validate, select Neural Network from the Regression drop down menu and choose Validate. The *Parameters* dialog box for Neural Network Validation will appear. In addition to the parameters explained above for Neural Network training, the following validation specific parameters need to be specified.

Number of Folds If N-Fold is chosen, specify the number of folds. The default is 3.

Number of Repeats The default is 1.

The results of validation with Neural Network are displayed in the navigator. The Neural Network view appears under the current spreadsheet and the results of validation are listed under it. They consist of the Regression Report and Statistical Report described below:

- Regression Report

  The report table gives the identifiers; the true value, the mean and standard deviation of predicted values across all repeats. The report can either be saved to an ASCII text file, or the *Predicted Value* and *Residual* columns can be exported back to the dataset.

- Statistical Report This report gives the mean absolute error, maximum absolute error and Root-Mean-Squared error for mean predicted values. It also report R2 computed on the mean predicted values.

14.8 Prediction

This section describes the Linear regression and Neural Networks prediction algorithms.

14.8.1 Linear Regression Predict

To predict with the Linear Regression algorithm, from the Regression drop down menu select Predict. The *Parameters* dialog box for Predict will appear. Browse to select the previously saved model file with extension .mdl, which is the result of training the linear regression with a dataset. Then click OK to execute. The results of regression with Linear Regression are displayed in the navigator. The Linear Regression view appears under the current spreadsheet and the results of regression are listed under it. These consist of the following views:

- Regression Report

  The report table gives the identifiers; the true value, and confidence for the prediction. The report can either be saved to an ASCII text file, or the *Predicted Value* and *Residual* columns can be exported back to the dataset.

14.8.2 Neural Network Predict

To predict with the Neural Network algorithm, from the Regression drop down menu select Predict. The *Parameters* dialog box for Predict will appear. Browse to select the previously saved model file with extension .mdl, which is the result of training the neural network with a dataset. Then click OK to execute. The results of regression with Neural Network are displayed in the navigator. The Neural Network view appears under the current spreadsheet and the results of regression are listed under it. These consist of the following views:

- Regression Report

  The report table gives the identifiers; and the true value. The report can either be saved to an ASCII text file, or the *Predicted Value* and *Residual* columns can be exported back to the dataset.

Chapter 15

Principal Component Analysis

15.1 Viewing Data Separation using Principal Component Analysis

Imagine trying to visualize the separation between various tumor types given gene expression data for several thousand genes for each sample. There is often sufficient redundancy in these large collection of genes and this fact can be used to some advantage in order to reduce the dimensionality of the input data. Visualizing data in 2 or 3 dimensions is much easier than doing so in higher dimensions and the aim of dimensionality reduction is to effectively reduce the number of dimensions to 2 or 3. There are two ways of doing this - either less important dimensions get dropped or several dimensions get combined to yield a smaller number of dimensions. The Principal Components Analysis (PCA) essentially does the latter by taking linear combinations of dimensions. Each linear combination is in fact an Eigen Vector of the similarity matrix associated with the dataset. These linear combinations (called Principal Axes) are ordered in decreasing order of associated Eigen Value. Typically, two or three of the top few linear combinations in this ordering serve as very good set of dimensions to project and view the data in. These dimensions capture most of the information in the data.

ArrayAssist supports a fast PCA implementation along with an interactive 2D viewer for the projected points in the smaller dimensional space. It clearly brings out the separation between different groups of rows/columns whenever such separations exist.

Note: Select *Statistics* → *PCA* from the menubar to initiate PCA. The following options are available when running PCA.

PCA on rows/columns option Use this option to indicate whether the PCA algorithm needs to be run on the rows or the columns of the dataset.

Specify a pruning option Typically, only the first few eigen-vectors (principal components) capture most of the variation in the data. The execution speed of PCA algorithm can be greatly enhanced when only a few eigenvectors are computed as compared to all. The pruning option determines how many eigenvectors are computed eventually. You can explicitly specify the exact number by selecting *Number of Principal Components* option, or specify that the algorithm compute as many eigenvectors as required to capture the specified *Total Percentage Variation* in the data.

Normalization Options Use this if the range of values in the data columns varies widely. These options normalize all columns to zero mean and unit standard deviation before performing PCA. This is enabled by default.

3D Plot The default output plot of PCA will be a 2D plot. If a 3D plot is desired in addition, then check this option.

15.2 Outputs of Principal Components Analysis

The output of PCA is shown in the following three views:

15.2.1 Principal Eigen Values

This is a plot of the Eigen values (E0, E1, E2, etc.) associated with the principal axes against their respective percentage contribution. The minimum number of principal axes required to capture most of the information in the data can be gauged from this plot. The blue line indicates the actual variation captured by each eigen-value, and the red line indicates the cumulative variation captured by all eigen values up to that point.

15.2.2 PCA Scores

This is a scatter plot of data projected along the principal axes (eigenvectors). By default, the first and second principal axes are plotted to begin

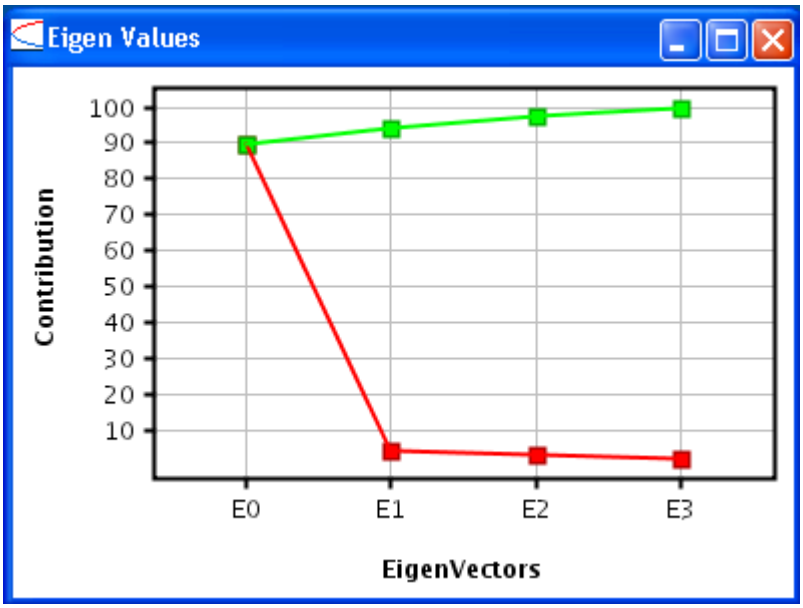


Figure 15.1: Eigen Value Plot

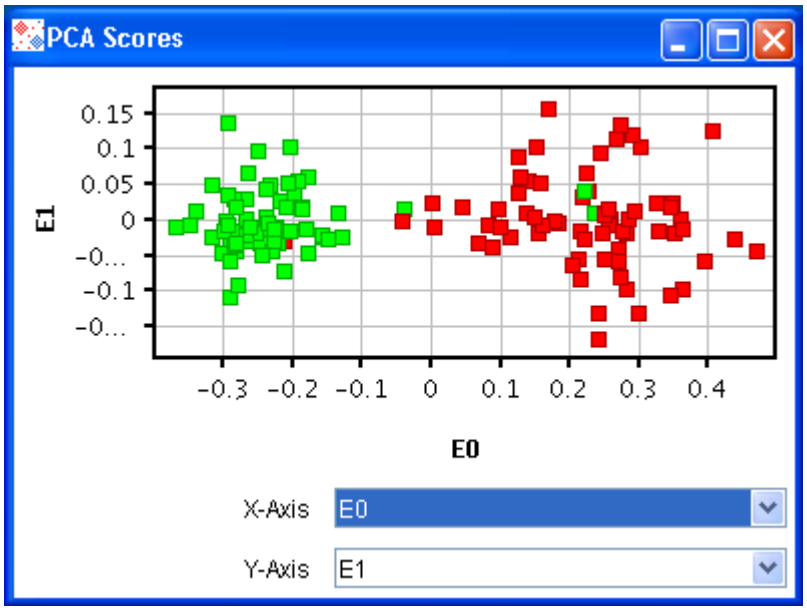


Figure 15.2: Scatter Plot of PCA Scores with multi-class data

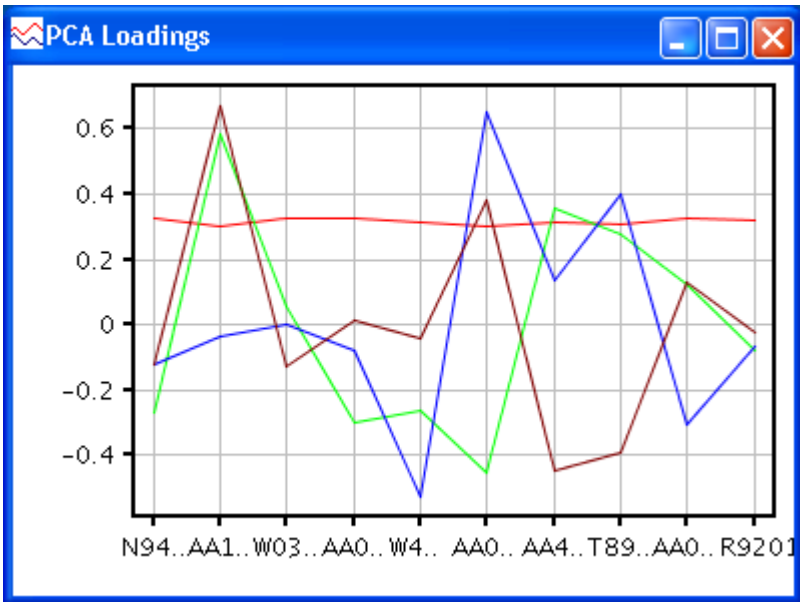


Figure 15.3: Scatter Plot of PCA Loadings

with, which capture the maximum variation of the data. If the dataset has a classlabel column, the points are colored w.r.t that column, and it is possible to visualize the separation (if any) of classes in the data. Different principal axes can be chosen using the dropdown menu for the *X-Axis* and *Y-Axis*. Each axes is labelled by its eigenvalue (i.e the percentage contribution to the total variation).

This view is a lassoed view and supports all operations and customizations like the *Scatter Plot* view.

In addition, the actual numerical scores can be saved to a tab-separated ASCII text file using the *Export As* → *Text* option in the right-click context menu. This data can then be loaded back into **ArrayAssist** for further analysis.

If the 3D option was exercised, then a similar 3D scores plot will also be shown with the top 3 principal components as the three axes.

15.2.3 PCA Loadings

As mentioned earlier, each principal component (or eigenvector) is a linear combination of the selected columns. The relative contribution of each

column to an eigenvector is called its *loading* and is depicted in the *PCA Loadings* plot. The X-Axis consists of columns, and the Y-Axis denotes the weight contributed to an eigenvector by that column. Each eigenvector is plotted as a profile, and it is possible to visualize whether there is a certain subset of columns which overwhelmingly contribute (large absolute value of weight) to an important eigenvector; this would indicate that those columns are important distinguishing features in the whole data.

A dropdown combo box indicates the eigenvalue associated with the current eigenvector (highlighted in yellow). Highlight the appropriate eigen vector using this combobox to inspect the relative contribution of columns to the selected eigenvector.

The actual numerical loadings can be saved to a tab-separated ASCII text file using the *Export As* $\rightarrow$ *Text* option in the right-click context menu. This data can then be loaded back into **ArrayAssist** for further analysis.

Chapter 16

Statistical Hypothesis Testing and Differential Expression Analysis

This chapter describes techniques available in **ArrayAssist** for Statistical Hypothesis Testing.

16.1 Differential Expression Analysis

The *Differential Expression Analysis* module in **ArrayAssist** analyses replicate experiments using statistical hypothesis testing algorithms to find statistical significance p-values and fold-changes for genes. (In case there are no replicates, only a fold-change will be computed).

Several different types of experiment designs can be handled by this module. Typical examples of situations where you can use this module to determine differentially expressed genes include the following, among others:

- You have run two groups of replicate experiments, say a control and a treatment group, and you wish to determine genes that are differentially expressed between control and treatment.
- You have run two or more groups of experiments, and you wish to determine genes which show significantly different behavior between groups or between any pair of groups.

For each of the above experiment types, appropriate statistical tests in **ArrayAssist** will determine significance p-values for each gene and also

fold-changes for each gene between pairs of groups.

16.1.1 The Differential Expression Analysis Wizard

Note that the structuring of data is such that columns in the data set correspond to experiments and rows to genes or spots. The *Differential Expression Wizard* assumes that **Experiment Grouping** has been performed on the data and that some **Probe Summarization** algorithm has been run on it. If any of these operations has not been performed already, one can do so now from the *WorkFlow Browser*. Each of the statistical tests described below will output p-values (and other auxiliary information) alongwith volcano plots. The Differential Expression Analysis Wizard is launched from *Statistics⟶Differential Expression Analysis*.

1. First step in the wizard involves setting the **Experiment Design**. Select the **Experiment Factors** and groups within factors to be considered for analysis. The interface (figure) shows a list of all the factors and groups available.

   The ensuing statistical tests have two versions, the **Unpaired** version and the **Paired** version. Use the unpaired version if the groups are derived from different sources or individuals. For instance, suppose one set of mice is subject to a certain treatment and another distinct set is taken for control, then use the unpaired option.

   Use the paired version if the same individuals are involved in the two groups at hand. For instance, suppose you take samples from a set of individuals, split these samples into two parts, use one part as control and treat the other part, then testing between control and treatment must be done with a paired test because control and treatment samples were derived from the same source.

   If the paired option is chosen then additionally one may have to do some **Column Reordering**, in the next step of the wizard, and pair up the corresponding replicates (figure).

2. In the next step of the wizard, select the **Analysis Type** (figure).

   - If you have only two groups or you have more than two groups but would like to compare groups pairwise, then use the **Analysis Type: Pairwise** option; this will allow you to determine differential expression between one or more pairs of groups simultaneously and also the p-values and fold-changes. Further, either

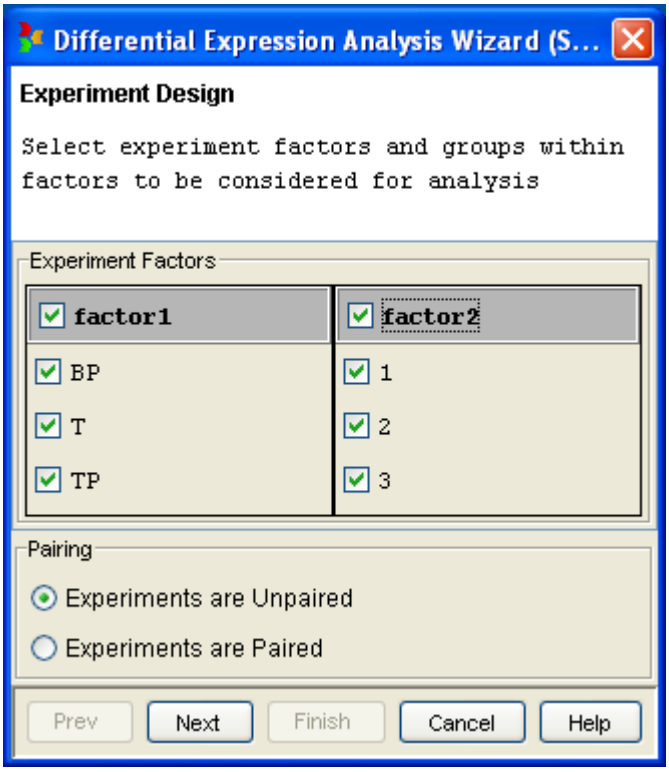


Figure 16.1: Experiment Design

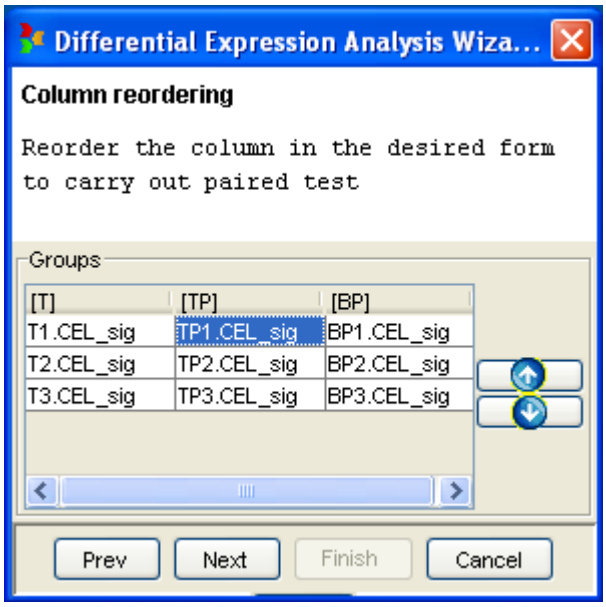


Figure 16.2: Column Reordering

you could choose to do calculations for selected pairs of groups or compare all groups with a reference group, in which case you would have to set the reference group.

- Alternatively, if you have more than two groups and would like to ask questions like "is the gene at hand differentially expressed in any of the groups" rather than "is it differentially expressed between a given pair of groups", use the **Analysis Type: All Together** option. For instance, if you have several replicates each of three or more treatments, choosing this option will perform statistical tests on genes which will indicate whether at least one of the treatments has a differential effect with respect to the other treatments. This option will compute a p-value for each gene (and no fold change).

3. The next step of the wizard is **Test Selection** (figure). Choose the appropriate test. Together, the analysis options, test type, and test options will determine the exact statistical test used for analysis. A list of statistical tests appears in the table below (table 1.1). Technical details of these steps are also described below.

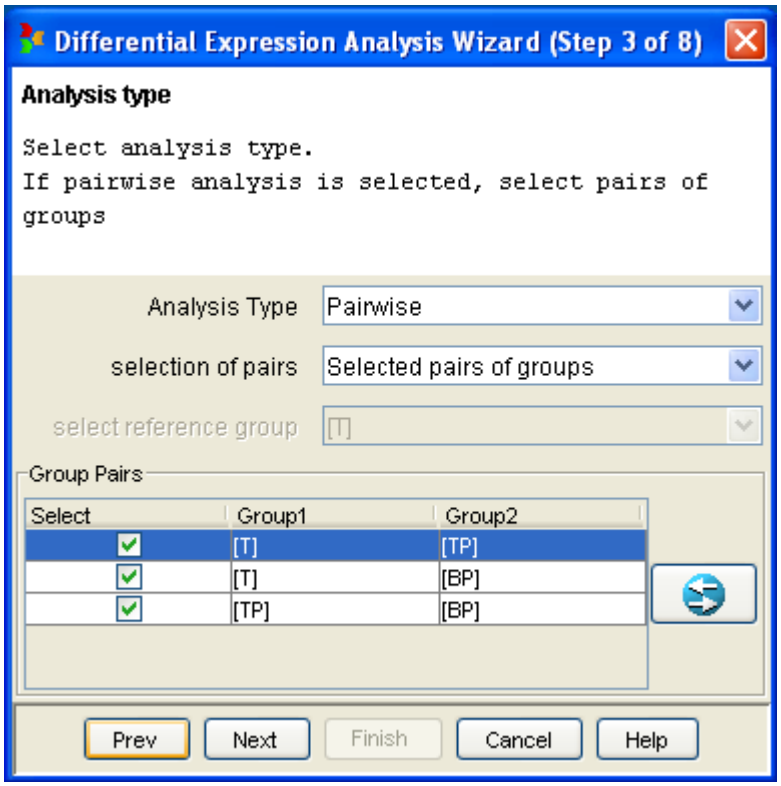


Figure 16.3: Analysis Type

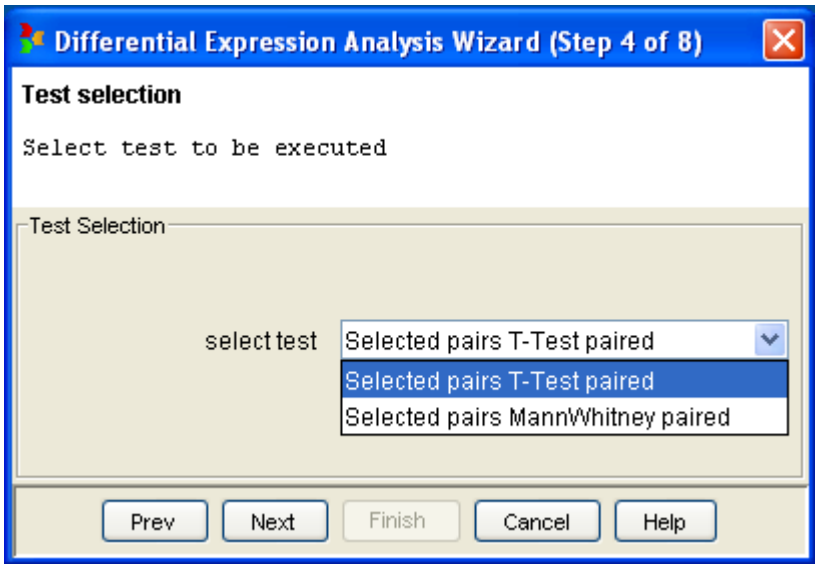


Figure 16.4: Select Test

The test type is either **Parametric** or **Non-Parametric**. Parametric analysis for a gene assumes that its expression values over various experiments are distributed normally. When this cannot be assumed, tests based on ranks, rather than actual values, are often more reliable and powerful. Such tests are called non-parametric tests. The parametric test option is the default.

The test options available are detailed below. Each of these tests will output a p-value for each gene.

- If a **Single Group** was chosen earlier, then the only test option available in this step is the t-Test against 0 for the parametric case and Mann Whitney against 0 for the non-parametric case.
- If **Two Groups** were chosen earlier, then the test option available in this step is the t-Test for the parametric case and Mann Whitney for the non-parametric case.
- If **More than Two Groups** were chosen earlier, and if **Pairwise** was chosen for **Selected Pairs of Groups** or **All Groups with a Reference Group** then the test option available in this step is the t-Test for the parametric case and Mann Whitney for the non-parametric case.

Analysis Type	Parametric	Non-Parametric
Single Group	t-Test against 0	Mann Whitney against 0
Multiple Groups, Unpaired, Pairwise Analysis	t-Test, Unpaired	Mann Whitney, Unpaired
Multiple Groups, Paired, Pairwise Analysis	t-Test, Paired	Mann Whitney, Paired
Multiple Groups, Unpaired, All Together	One-Way ANOVA	Kruskal-Wallis
Multiple Groups, Paired, All Together	Repeated Measures	Repeated Measures (Friedman)
Multiple Factors, Multiple Groups, Unpaired, All Together	n-Way ANOVA	None
Multiple Factors, Multiple Groups, Paired, All Together	Repeated Measures	None

Table 16.1: Table of Statistical Tests supported in **ArrayAssist**

However, if **All Together** was chosen then the test option available is ANOVA for the parametric case and Kruskal-Wallis for the non-parametric case.

- If **Multiple Factors** were chosen, wherein the same number of individuals appear in all the factors, under various groups, then an n-way ANOVA test is available for the *Unpaired* case while Repeated Measures test is available for the *Paired* case.

  Say, a certain collection of individuals are observed over time, for the effect of some drug *versus* pacebo, with multiple factors like age, sex, body weight, drug dosage etc. influencing the results. In such a case, one would have to run the above mentioned tests to measure the effect of various factors over the results.

  Note that the *Paired* option would be valid only if the various factors and groups are balanced, i.e., groups and experiment factors selected for analysis have equal number of observations. Suppose some experiments were carried out on male and female rats with two doses of medicine. Now, if one wants to carry out paired analysis with all the factors considered, then it is necessary to have same number of observations in the following categories: male–dose-1, male–dose-2, female–dose1, female–dose-2.

Technical descriptions of these tests appear later in the chapter.

4. The last step of the wizard is **P-value Computation** (figure). Each of the above tests will return a p-value for each gene. This p-value can either be computed using **Asymptotic** analysis or **Permutative** analysis. The former option computes p-values based on the assumption that the distribution is normal while the latter option does not rely on this assumption. The permutative analysis method is available only for the unpaired t-Test, the unpaired Mann-Whitney test and the One-Way ANOVA test.

   Also, select the **Multiple Testing Correction** method to get a corrected p-value. Choose one of the following correction algorithms: **Bonferroni Holm FWER**, **Westfall Young Permutative** or **Benjamini Hochberg FDR**. Alternatively, you can choose to have **No Correction**, in which case the original p-values will be retained. Note that the Westfall Young Permutative option is not available for *paired* tests.

   Technical details on how these methods work and why correction is needed, are detailed later in this chapter. Note, however, that correction methods are often too conservative, i.e., they err too much on the side of caution in determining significance of differential expression.

   Note: We have implemented a batch processing mode for significance analysis computations, for handling datasets with a very large number of rows. The batch size parameter can be set by the Tools ⟶Options ⟶Statistics. The default batch size is set to 30000. However, the permutative p-value computation as well as the Westfall Young permutative multiple testing correction requires that the whole dataset be loaded into memory for doing the computations. If the number of rows in the dataset is very large; larger than twice the batch size; then the permutative p-value computation and the Westfall Young permutative multiple testing correction will not be available. If you increase the batch size to a very high value the algorithm may be slow.

5. Processing begins now and **ArrayAssist** comes up with a spreadsheet with various calculated values (figure) and a report which shows a table containing the number of genes satisfying various p-value and fold-change combinations (figure 1.7). Also, a volcano plot is displayed which is a plot between log of fold-change and log of p-value (figure).

   For the case of single groups or multiple groups analyzed all together, fold-changes will not be computed and only a p-value table will be

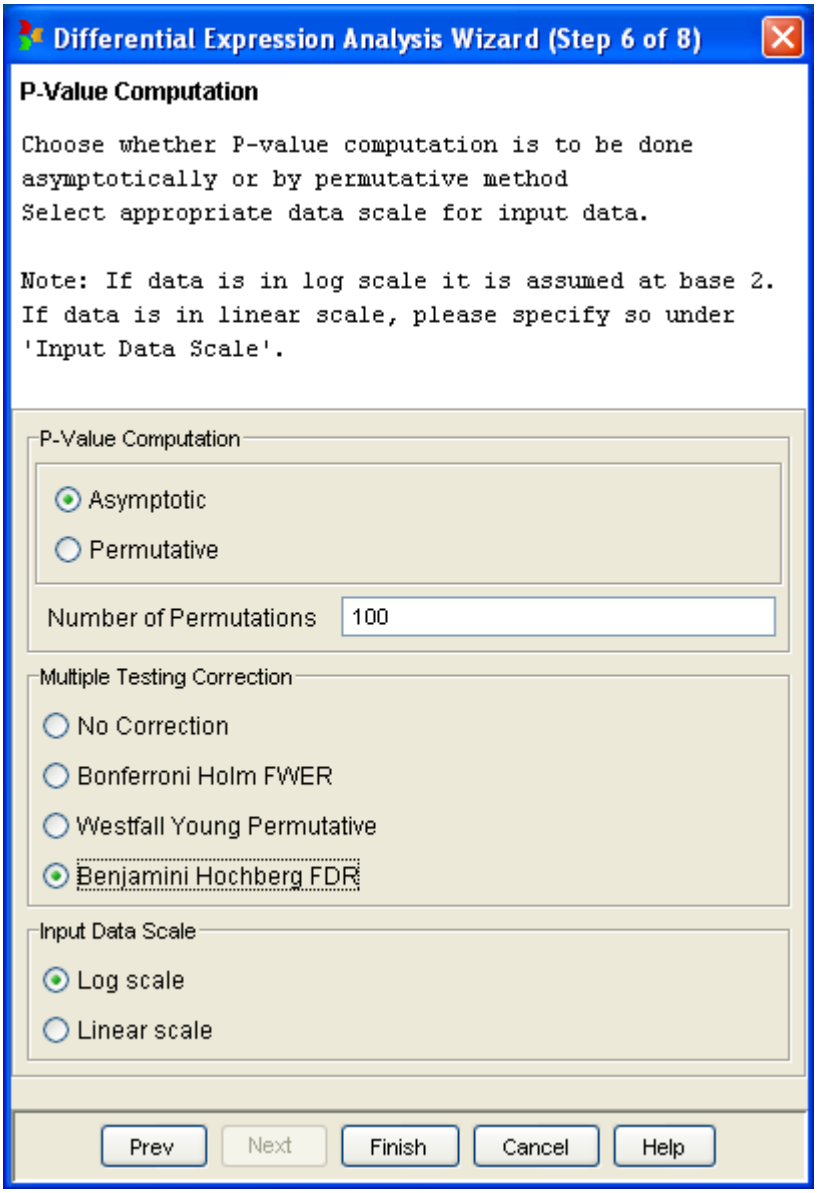
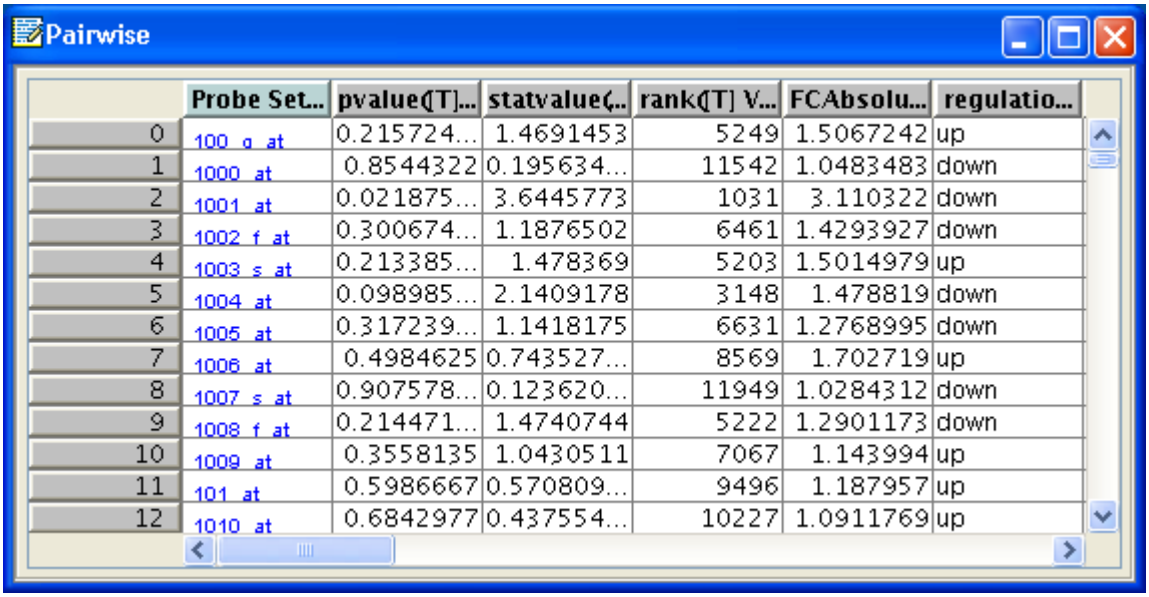


Figure 16.5: P-value Computation



	Probe Set...	pvalue(T]...	statvalue(...	rank(T] V...	FCAbsolu...	regulatio...
0	100_g_at	0.215724...	1.4691453	5249	1.5067242	up
1	1000_at	0.8544322	0.195634...	11542	1.0483483	down
2	1001_at	0.021875...	3.6445773	1031	3.110322	down
3	1002_f_at	0.300674...	1.1876502	6461	1.4293927	down
4	1003_s_at	0.213385...	1.478369	5203	1.5014979	up
5	1004_at	0.098985...	2.1409178	3148	1.478819	down
6	1005_at	0.317239...	1.1418175	6631	1.2768995	down
7	1006_at	0.4984625	0.743527...	8569	1.702719	up
8	1007_s_at	0.907578...	0.123620...	11949	1.0284312	down
9	1008_f_at	0.214471...	1.4740744	5222	1.2901173	down
10	1009_at	0.3558135	1.0430511	7067	1.143994	up
11	101_at	0.5986667	0.570809...	9496	1.187957	up
12	1010_at	0.6842977	0.437554...	10227	1.0911769	up

Figure 16.6: Differential Expression Spread-sheet

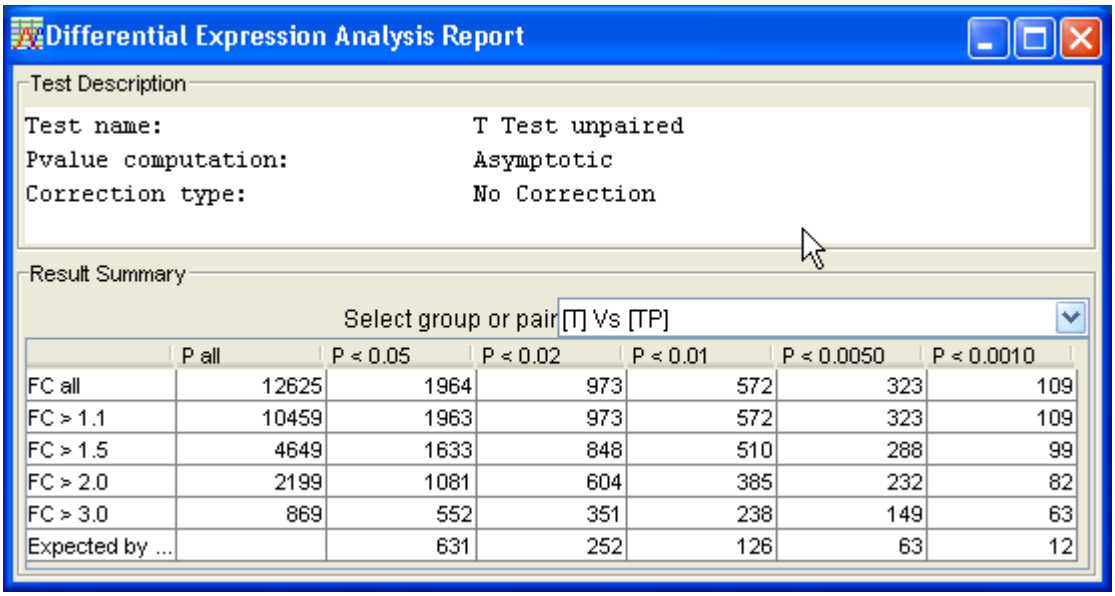


Figure 16.7: Differential Expression Analysis Report

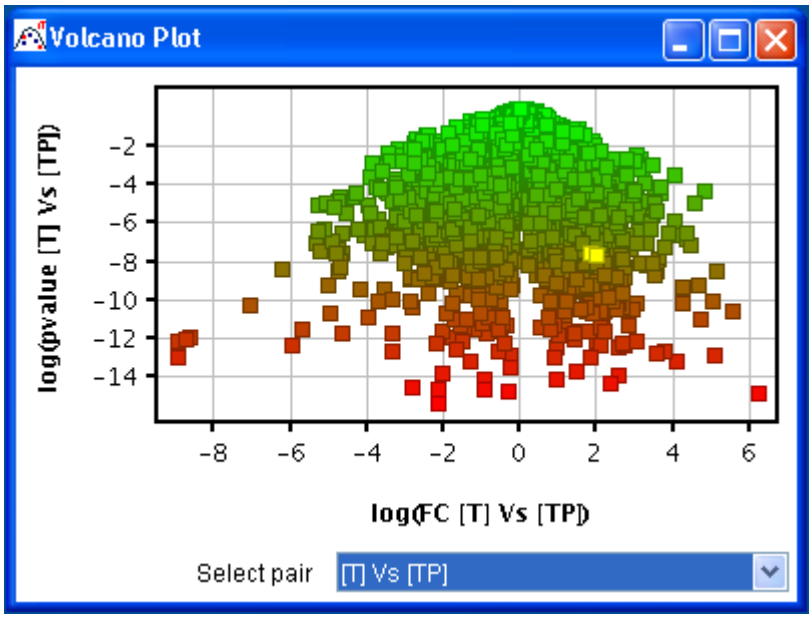


Figure 16.8: Volcano Plot

shown. Thus, the volcano plot will also not be displayed. Further, for the multiple groups pairwise analysis option, there may be multiple tables created and these can be accessed through the drop-down list in the Differential Expression Analysis Report view. Same holds true for the Volcano Plot.

16.2 Analyzing Non-Replicate Data

If you have non-replicate data and would like to analyze this, then the differential expression module will not be totally applicable. If you just have a group with no replicates, there is no analysis that can be done. In case of two groups, without replicates, one can compute a fold-change with respect to one of the groups taken as reference. With more than two groups, without replicates, one can look at the fold-change in all the groups with respect to a reference group.

Note that in absence of replicates, p-value computation and related multiple testing correction is not possible.

16.3 Technical Details of Replicate Analysis

Replicate analysis to determine differential expression across groups is performed using what is called *statistical hypothesis testing*. To explain the need for statistical hypothesis testing, as opposed to simple measures like fold-changes, consider the simple case of two groups of experiments, typically a control group and a treatment group, with each group having several replicates. The fold-change measure computes the difference between the group means for each gene. A cut-off on this quantity is then used to determine genes which are differentially expressed. However, this gives a very large number of false positives. This stems from the fact that most genes are expressed at low levels where the signal to noise ratio is low and therefore fold changes occur at random for a large number of genes. Further, at high expression levels, small but consistent changes in expression across experiments are not detected by fold-change. Statistical hypothesis testing offers a better alternative.

16.3.1 Statistical Tests

A brief description of the various statistical tests in **ArrayAssist** appears below. See [26] for a simple introduction to these tests.

The Unpaired t-Test for Two Groups: The standard test that is performed in such situations is the so called t-test, which measures the following t-statistic for each gene g (see, e.g., [26]):

$$t_g = \frac{m_1 - m_2}{\sqrt{s_1^2/n_1 + s_2^2/n_2}}$$

Here, m_1, m_2 are the mean expression values for gene g within groups 1 and 2, respectively, s_1, s_2 are the corresponding standard deviations, and n_1, n_2 are the number of experiments in the two groups. Qualitatively, this t-statistic has a high absolute value for a gene if the means within the two sets of replicates are very different and if each set of replicates has small standard deviation. Thus, the higher the t-statistic is in absolute value, the greater the confidence with which this gene can be declared as being differentially expressed. Note that this is a more sophisticated measure than the commonly used fold-change measure (which would just be $m_1 - m_2$ on the log-scale) in that it looks for a large fold-change in conjunction with small variances in each group, The power of this statistic in differentiating between true differential expression and differential expression due to random effects increases as the numbers n_1 and n_2 increase.

The t-Test against 0 for a Single Group: This is performed on one group using the formula

$$t_g = \frac{m_1}{\sqrt{s_1^2/n_1}}$$

The Paired t-Test for Two Groups: The paired t-test is done in two steps. Let $a_1 \ldots a_n$ be the values for gene g in the first group and $b_1 \ldots b_n$ be the values for gene g in the second group.

- First, the paired items in the two groups are subtracted, i.e., $a_i - b_i$ is computed for all i.
- A t-test against 0 is performed on this single group of $a_i - b_i$ values.

The Unpaired Mann-Whitney Test: The t-Test assumes that the gene expression values within groups 1 and 2 are independently and randomly drawn from the source population **and** obey a normal distribution. If the latter assumption may not be reasonably supposed, the preferred test is the non-parametric Mann-Whitney test, sometimes referred to as the Wilcoxon Rank-Sum test. It only assumes that the data within a sample are obtained from the same distribution but requires no knowledge of that distribution. The test combines the raw data from the two samples of size n_1 and n_2 respectively into a single sample of size $n = n_1 + n_2$. It then sorts the data and provides ranks based on the sorted values. Ties are resolved by giving averaged values for ranks. The data thus ranked is returned to the original sample group 1 or 2. All further manipulations of data are now performed on the rank values rather than the raw data values. The probability of erroneously concluding differential expression is dictated by the distribution of T_i, the sum of ranks for group i, $i = 1, 2$. This distribution can be shown to be normal mean $m_i = n_i(\frac{n+1}{2})$ and standard deviation $\sigma_1 = \sigma_2 = \sigma$, where σ is the standard deviation of the combined sample set.

The Paired Mann-Whitney Test: The samples being paired, the test requires that the sample size of groups 1 and 2 be equal, i.e., $n_1 = n_2$. The absolute value of the difference between the paired samples is computed and then ranked in increasing order, apportioning tied ranks when necessary. The statistic T, representing the sum of the ranks of the absolute differences taking non-zero values obeys a normal distribution with mean $m = \frac{1}{2}(n_1\frac{(n_1+1)}{2}) - S_0)$, where S_0 is the sum of the ranks of the differences taking value 0, and variance given by one-fourth the sum of the squares of the ranks.

The Mann-Whitney and t-test described previously address the analysis of two groups of data; in case of three or more groups, the following tests may be used.

One-Way ANOVA: When comparing data across three or more groups, the obvious option of considering data one pair at a time presents itself. The problem with this approach is that it does not allow one to draw any conclusions about the dataset as a whole. While the probability that each individual pair yields significant results by mere chance is small, the probability that any one pair of the entire dataset does so is substantially larger. The One-Way ANOVA takes a comprehensive approach in analyzing data and attempts to extend the logic of t-tests to handle three or more groups concurrently. It uses the mean of the sum of squared deviates (SSD) as an aggregate measure of variability between and within groups. NOTE: For a sample of n observations $X_1, X_2, ...X_n$, the sum of squared deviates is given by

$$SSD = \sum_{i=1}^{n} X_i^2 - \frac{(\sum_{i=1}^{n} X_i)^2}{n}$$

The numerator in the t-statistic is representative of the difference in the mean **between** the two groups under scrutiny, while the denominator is a measure of the random variance **within** each group. For a dataset with k groups of size $n_1, n_2, ...n_k$, and mean values $M_1, M_2, ..., M_k$ respectively, One-Way ANOVA employs the SSD between groups, SSD_{bg}, as a measure of variability in group mean values, and the SSD within groups, SSD_{wg} as representative of the randomness of values within groups. Here,

$$SSD_{bg} \equiv \sum_{i=1}^{k} n_i (M_i - M)^2$$

and

$$SSD_{wg} \equiv \sum_{i=1}^{k} SSD_i$$

with M being the average value over the entire dataset and SSD_i the SSD within group i. (Of course it follows that sum $SSD_{bg} + SSD_{wg}$ is exactly the total variability of the entire data).

Again drawing a parallel to the t-test, computation of the variance is associated with the number of degrees of freedom (df) within the sample,

which as seen earlier is $n-1$ in the case of an n-sized sample. One might then reasonably suppose that SSD_{bg} has $df_{bg} = k-1$ degrees of freedom and SSD_{wg}, $df_{wg} = \sum_{i=1}^{k} n_i - 1$. The mean of the squared deviates (MSD) in each case provides a measure of the variance between and within groups respectively and is given by $MSD_{bg} = \frac{SSD_{bg}}{df_{bg}}$ and $MSD_{wg} = \frac{SSD_{wg}}{df_{wg}}$.

If the null hypothesis is false, then one would expect the variability between groups to be substantial in comparison to that within groups. Thus MSD_{bg} may be thought of in some sense as $MSD_{hypothesis}$ and MSD_{wg} as MSD_{random}. This evaluation is formalized through computation of the

$$F - ratio = \frac{MSD_{bg}/df_{bg}}{MSD_{wg}/df_{wg}}$$

It can be shown that the F-ratio obeys the F-distribution with degrees of freedom df_{bg}, df_{wg}; thus p-values may be easily assigned.

The One-Way ANOVA assumes independent and random samples drawn from a normally distributed source. Additionally, it also assumes that the groups have approximately equal variances, which can be practically enforced by requiring the ratio of the largest to the smallest group variance to fall below a factor of 1.5. These assumptions are especially important in case of unequal group-sizes. When group-sizes **are** equal, the test is amazingly robust, and holds well even when the underlying source distribution is not normal, as long as the samples are independent and random. In the unfortunate circumstance that the assumptions stated above do not hold and the group sizes are perversely unequal, we turn to the Kruskal-Wallis test.

The Kruskal-Wallis Test: The Kruskal-Wallis (KW) test is the non-parametric alternative to the One-Way independent samples ANOVA, and is in fact often considered to be performing "ANOVA by rank". The preliminaries for the KW test follow the Mann-Whitney procedure almost verbatim. Data from the k groups to be analyzed are combined into a single set, sorted, ranked and then returned to the original group. All further analysis is performed on the returned ranks rather than the raw data. Now, departing from the Mann-Whitney algorithm, the KW test computes the **mean** (instead of simply the sum) of the ranks for each group, as well as over the entire dataset. As in One-Way ANOVA, the sum of squared deviates between groups, SSD_{bg}, is used as a metric for the degree to which group means differ. As before, the understanding is that the groups means will not differ substantially in case of the null hypothesis. For a dataset with k

groups of sizes $n_1, n_2, ..., n_k$ each, $n = \sum_{i=1}^{k} n_i$ ranks will be accorded. Generally speaking, apportioning these n ranks amongst the k groups is simply a problem in combinatorics. Of course SSD_{bg} will assume a different value for each permutation/assignment of ranks. It can be shown that the mean value for SSD_{bg} over all permutations is $(k-1)\frac{n(n-1)}{12}$. Normalizing the observed SSD_{bg} with this mean value gives us the H-ratio, and a rigorous method for assessment of associated p-values: The distribution of the

$$H - ratio = \frac{SSD_{bg}}{\frac{n(n+1)}{12}}$$

may be neatly approximated by the chi-squared distribution with $k-1$ degrees of freedom.

The Repeated Measures ANOVA: Two groups of data with inherent correlations may be analyzed via the paired t-Test and Mann-Whitney. For three or more groups, the Repeated Measures ANOVA (RMA) test is used. The RMA test is a close cousin of the basic, simple One-Way independent samples ANOVA, in that it treads the same path, using the sum of squared deviates as a measure of variability between and within groups. However, it also takes additional steps to effectively remove extraneous sources of variability, that originate in pre-existing individual differences. This manifests in a third sum of squared deviates that is computed for each individual set or row of observations. In a dataset with k groups, each of size n,

$$SSD_{ind} = \sum_{i=1}^{n} k(A_i - M)^2$$

where M is the sample mean, averaged over the entire dataset and A_i is the mean of the $k$values taken by individual/row i. The computation of SSD_{ind} is similar to that of SSD_{bg}, except that values are averaged over individuals or rows rather than groups. The SSD_{ind} thus reflects the difference in mean per individual from the collective mean, and has $df_{ind} = n - 1$ degrees of freedom. This component is removed from the variability seen within groups, leaving behind fluctuations due to "true" random variance. The F-ratio, is still defined as $\frac{MSD_{hypothesis}}{MSD_{random}}$, but while $MSD_{hypothesis} = MSD_{bg} = \frac{SSD_{bg}}{df_{bg}}$ as in the garden-variety ANOVA.

$$MSD_{random} = \frac{SSD_{wg} - SSD_{ind}}{df_{wg} - df_{ind}}$$

Computation of p-values follows as before, from the F-distribution, with degrees of freedom $df_{bg}, df_{wg} - df_{ind}$.

The Repeated Measures Friedman Test: As has been mentioned before, ANOVA is a robust technique and may be used under fairly general conditions, provided that the groups being assessed are of the same size. The non-parametric Kruskal Wallis test is used to analyst independent data when group-sizes are unequal. In case of correlated data however, group-sizes are necessarily equal. What then is the relevance of the Friedman test and when is it applicable? The Friedman test may be employed when the data is collection of ranks or ratings, or alternately, when it is measured on a non-linear scale.

To begin with, data is sorted and ranked **for each individual or row** unlike in the Mann Whitney and Kruskal Wallis tests, where the entire dataset is bundled, sorted and then ranked. The remaining steps for the most part, mirror those in the Kruskal Wallis procedure. The sum of squared deviates between groups is calculated and converted into a measure quite like the H measure; the difference however, lies in the details of this operation. The numerator continues to be SSD_{bg}, but the denominator changes to $\frac{k(k+1)}{12}$, reflecting ranks accorded to each individual or row.

The Two-way ANOVA: The Two-Way ANOVA is used to determine the effect due to two parameters concurrently. It assesses the individual influence of each parameter, as well as their net interactive effect. Proceeding as in One-Way ANOVA, sum of squared deviates between and within groups SS_{bg} and SS_{wg} are calculated. The latter is used directly to compute MSD_{random}, while the former is split into three components:

$$SS_{bg} = SS_{parameter1} + SS_{parameter2} + SS_{interaction}$$

$SS_{parameter1}$ and $SS_{parameter2}$ are derived through the standard formula for computing sum of squared deviates. The associated number of degrees of freedom in each case and the ratios $MSD_{parameter1}$, $MSD_{parameter2}$, and $MSD_{interaction}$ are computed. The three MSDs when divided by MSD_{random} yield three F-ratios and associated p-values/tests of significance.

16.3.2 Obtaining P-Values

Each statistical test above will generate a test value or statistic called the *test metric* for each gene. Typically, larger the test-metric more significant the differential expression for the gene in question. To identify all differentially

expressed genes, one could just sort the genes by their respective test-metrics and then apply a cutoff. However, determining that cutoff value would be easier if the test-metric could be converted to a more intuitive p-value which gives the probability that the gene g appears as differentially expressed purely by chance. So a p-value of .01 would mean that there is a 1% chance that the gene is not really differentially expressed but random effects have conspired to make it look so. Clearly, the actual p-value for a particular gene will depend on how expression values within each set of replicates are distributed. These distributions may not always be known.

Under the assumption that the expression values for a gene within each group are normally distributed and that the variances of the normal distributions associated with the two groups are the same, the above computed test-metrics for each gene can be converted into p-values, in most cases using closed form expressions. This way of deriving p-values is called *Asymptotic* analysis. However, if you do not want to make the normality assumptions, a *permutation analysis* method is sometimes used as described below.

p-values via Permutation Tests: As described in Dudoit et al. [25], this method does not assume that the test-metrics computed follows a certain fixed distribution.

Imagine a spreadsheet with genes along the rows and arrays along columns, with the first n_1 columns belonging to the first group of replicates and the remaining n_2 columns belonging to the second group of replicates. The left to right order of the columns is now shuffled several times. In each trial, the first n_1 columns are treated as if they comprise the first group and the remaining n_2 columns are treated as if they comprise the second group; the t-statistic is now computed for each gene with this new grouping. This procedure is ideally repeated $\binom{n_1+n_2}{n_1}$ times, once for each way of grouping the columns into two groups of size n_1 and n_2, respectively. However, if this is too expensive computationally, a large enough number of random permutations are generated instead.

p-values for genes are now computed as follows. Recall that each gene has an actual test metric as computed a little earlier and several permutation test metrics computed above. For a particular gene, its p-value is the fraction of permutations in which the test metric computed is larger in absolute value than the actual test metric for that gene.

16.3.3 Adjusting for Multiple Comparisons

Microarrays usually have genes running into several thousands and tens of thousands. This leads to the following problem. Suppose p-values for each

gene have been computed as above and all genes with a p-value of less than .01 are considered. Let k be the number of such genes. Each of these genes has a less than 1 in 100 chance of appearing to be differentially expressed by random chance. However, the chance that *at least* one of these k genes appears differentially expressed by chance is much higher than 1 in 100 (as an analogy, consider fair coin tosses, each toss produces heads with a 1/2 chance, but the chance of getting at least one heads in a hundred tosses is much higher). In fact, this probability could be as high $k * .01$ (or in fact $1-(1-.01)^k$ if the p-values for these genes are assumed to be independently distributed). Thus, a p-value of .01 for k genes does not translate to a 99 in 100 chance of all these genes being truly differentially expressed; in fact, assuming so could lead to a large number of false positives. To be able to apply a p-value cut-off of .01 and claim that all the genes which pass this cut-off are indeed truly differentially expressed with a .99 probability, an adjustment needs to be made to these p-values.

See Dudoit et al. [25] and the book by Glantz [26] for detailed descriptions of various algorithms for adjusting the p-values. The simplest methods called the Holm step-down method and the Benjamini-Hochberg step-up methods are motivated by the description in the previous paragraph.

The Holm method: Genes are sorted in increasing order of p-value. The p-value of the jth gene in this order is now multiplied by $(n-j+1)$ to get the new adjusted p-value.

The Benjamini-Hochberg method: This method [24] assumes independence of p-values across genes, the p-value of the jth gene in the above order is multiplied by n/j, where n is the total number of genes (so the multiplier for gene 1 is n and for gene n is 1, as in the Holm step-down method).

In typical use, the former method usually turns out to be too conservative (i.e., the p-values end up too high even for truly differentially expressed genes) while the latter does not apply to situations where gene behavior is highly correlated, as is indeed the case in practice. Dudoit et al. [25] recommend the Westfall and Young procedure as a less conservative procedure which handles dependencies between genes.

The Westfall-Young method: The Westfall and Young [27] procedure is a permutation procedure in which genes are first sorted by increasing t-statistic obtained on unpermuted data. Then, for each permutation, the test metrics obtained for the various genes in this permutation are artificially adjusted so that the following property holds: if gene i has a higher original test-metric than gene j, then gene i has a higher adjusted test metric for this permutation than gene j. The overall corrected p-value for a gene is now

defined as the fraction of permutations in which the adjusted test metric for that permutation exceeds the test metric computed on the unpermuted data. Finally, an artificial adjustment is performed on the p-values so a gene with a higher unpermuted test metric has a lower p-value than a gene with a lower unpermuted test metric; this adjustment simply increases the p-value of the latter gene, if necessary, to make it equal to the former. Though not explicitly stated, a similar adjustment is usually performed with all other algorithms described here as well.

Chapter 17

ArrayAssist Enterprise Client

NOTE: You will need to have the enterprise client module of **ArrayAssist** to connect to the **Enterprise Server** and use the features available in this section.

The enterprise client module provides **ArrayAssist** the functionality to communicate with an **Enterprise Server**. This is distributed as a separate module with **ArrayAssist**. When the enterprise client module is activated, a new menu item appears on the top menu providing access to the **Enterprise Server**. Along with the Enterprise menu, an Enterprise tab appears along with the navigator tab on the left pane of the tool. The screenshot below the features of the client module that appear in **ArrayAssist**. The features of the client module that provide functionality for **ArrayAssist** to communicate with the **Enterprise Server** are detailed in this chapter.

The generic features of the **Enterprise Server** are outlined in the next section.

17.1 Enterprise Server

The **Enterprise Server** is a flexible and scalable system to be used with a range of client products. The Enterprise server is a generic server component that is meant to provide an enterprise wide functionality for storing and sharing data. The Enterprise server has the following features:

- Provides an enterprise wide data management system.

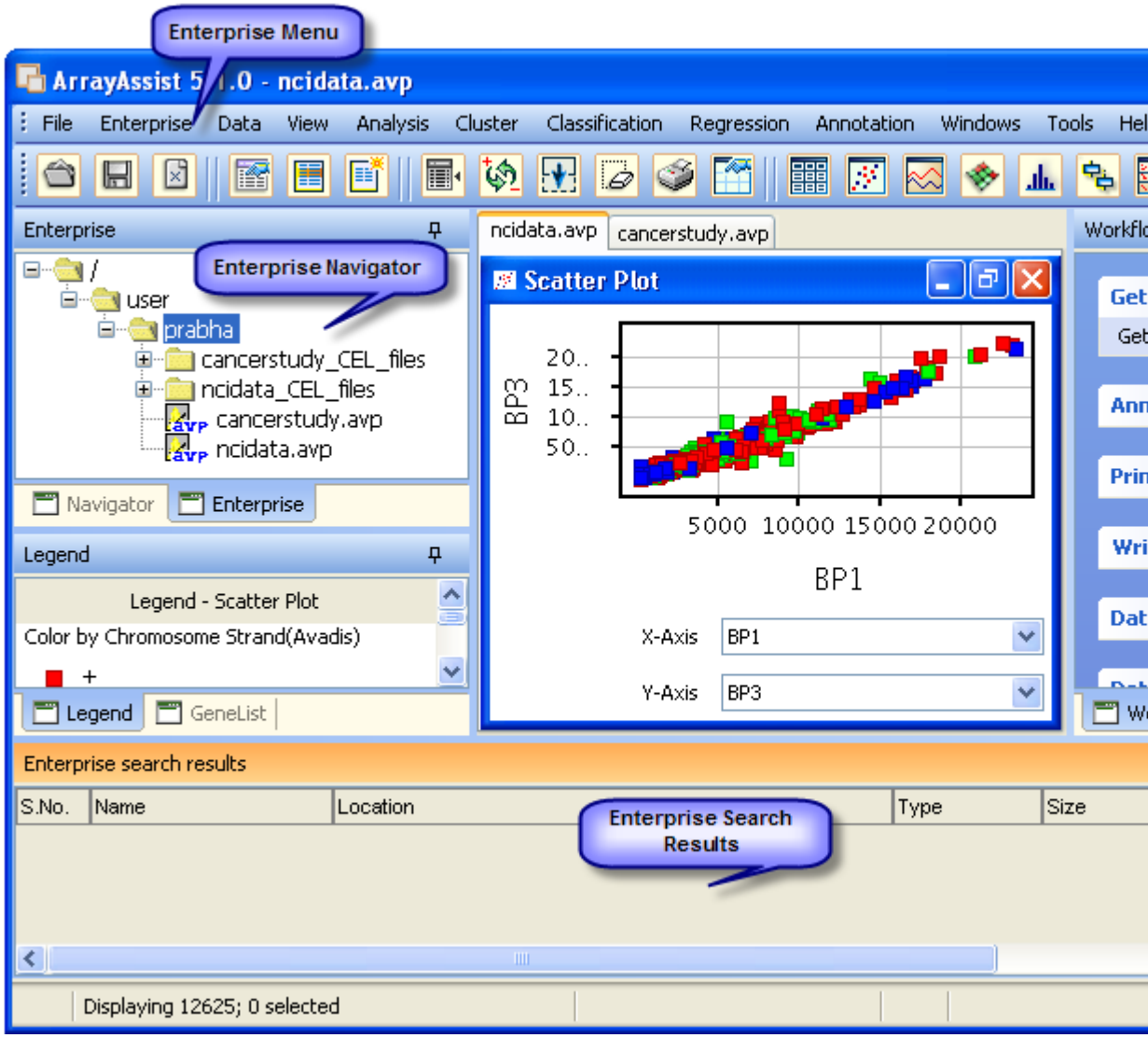


Figure 17.1: **ArrayAssist** Layout

- Provides user and group support with flexible access control.
- Provides full version control for all resources stored on the server.
- Supports secure communication between clients and server.
- Maintains access and data change logs.
- Supports full backup and restore functionality.
- Presents data in a hierarchical file structure.
- Support for associating meta data and annotation with every resources on the server that can be queried and searched.
- Provides user controlled automatic upload of resources to the server.
- Server infrastructure supports an independent Compute Server for running resource intensive algorithms, process integration and running custom workflows.
- Server infrastructure supports a synchronised field of Enterprise Servers.
- The **Enterprise Server** provides a rich application programing interface (API) that allows multiple clients and custom applications to access all the server functionality.

17.2 Setting up the Enterprise Server for ArrayAssist

NOTE: You will need to have administrative privileges for setting up the **Enterprise Server** for **ArrayAssist**

Before you start using the enterprise the **ArrayAssist** Enterprise Server the Administrator has to set up user accounts and user repositories on the **Enterprise Server** for all users. Details of setting these up are given in the **Enterprise Server** manual. In addition to setting up user accounts and repositories, the **Enterprise Server** administrator has to set up some libraries that will be used for all projects saved on the **Enterprise Server**. These libraries pertain to the vocabulary that will be used for the MIAME annotations.

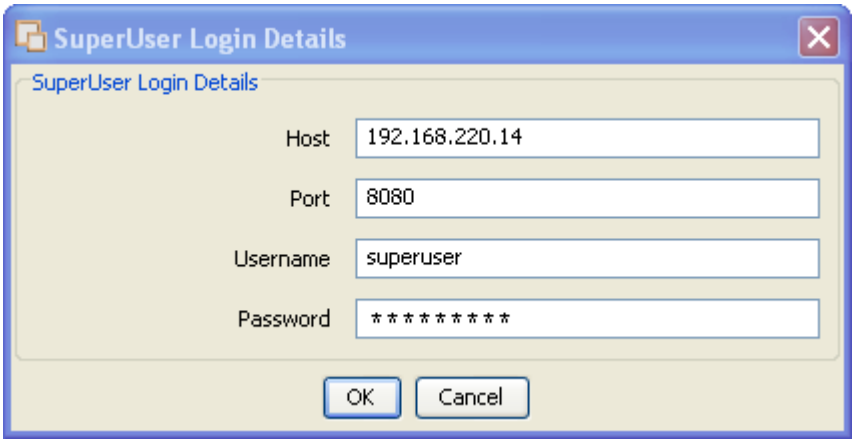


Figure 17.2: Superuser Login Details Dialog

17.2.1 Setting up Vocabularies for MIAME annotations

The **Enterprise Server** administrator needs to set up the vocabularies necessary for MIAME annotation. These vocabularies are packaged with **ArrayAssist** client module. To set up these vocabularies launch the **ArrayAssist** and open any sample project. Open the script editor, paste the following line into the script editor and click the Run icon button on the script editor.

```
script.enterpriseAdmin.createAAManager()
```

This will pop-up a dialog asking for the **Enterprise Server** and the superuser details. You will need by superuser to set the vocabularies. Enter the required details and click *OK*. This will prompt for repository details for the aamanager. This should normally be a subdirectory called aamanager under the main resource for enterprise data. For Example,

```
EnterpriseData\aamanager
```

The script will be executed to create an aamanager account on the **Enterprise Server**. It will then upload the vocabulary files that are required for MIAME annotations onto the server. These MIAME annotation files can then be used by all the projects on the **Enterprise Server**.

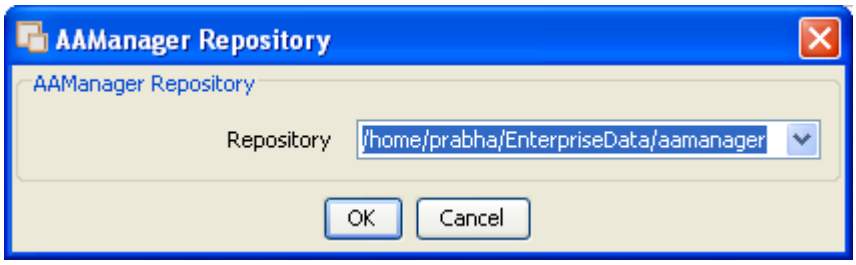


Figure 17.3: Array Assist Manager Repository setup

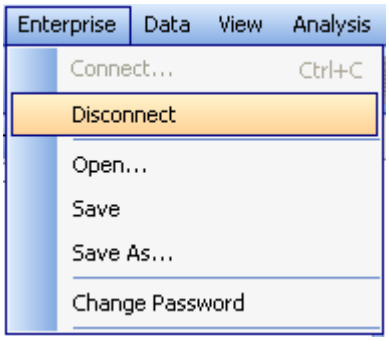


Figure 17.4: The Enterprise Menu on **ArrayAssist**

17.3 Logging in and Logging out of the Enterprise Server

If the Enterprise client module is available in the **ArrayAssist** client, an Enterprise menu item will appear on the menu bar of **ArrayAssist**. This has the menu items that allow you to connect and disconnect to the **Enterprise Server**; open and save projects from the **Enterprise Server**; and to change your password on the **Enterprise Server**.

17.3.1 Logging into the Enterprise Server

To connect to the **Enterprise Server**, choose *Enterprise* $\longrightarrow$*Connect* from the main menu on **ArrayAssist**. This will launch the connection dialog.

Enter the server details, user name and password and click *OK*. This will open a connection to the server and login to the **Enterprise Server** after authentication.

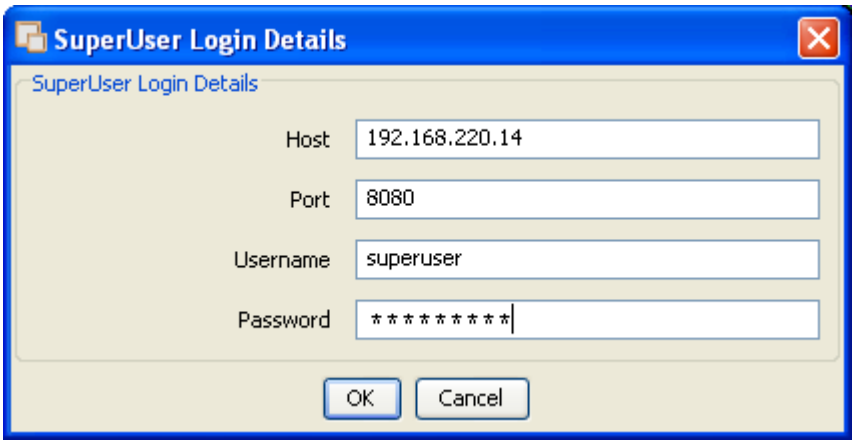


Figure 17.5: **Enterprise Server** Login Dialog for Creating aamanager

NOTE: If you want to login to the **Enterprise Server** through a proxy server, the proxy server details have to be provided in the *Tools* $\longrightarrow$*Options* $\longrightarrow$*Network Sittings* $\longrightarrow$*Proxy Settings*. These settings are global in the tool and will be used for all connections that **ArrayAssist** make with any other machine on the network.

After the connection to the **Enterprise Server** is established, the resources available in the **Enterprise Server** will be available in **ArrayAssist**. These will be shown as a tree in the *Enterprise* browser in the left panel of the tools as tab to the *Navigator* browser.

17.3.2 Change Password on the Enterprise Server

You can change your password on the **Enterprise Server** after you login. Go to the *Enterprise* $\longrightarrow$*Change Password* menu from **ArrayAssist** and change the password from the *Change Password Dialog*

17.3.3 Logging out from the Enterprise Server

To logout of the **Enterprise Server** use the *Enterprise* $\longrightarrow$*Disconnect* menu from the menu of **ArrayAssist**. This will log you out of the **Enterprise Server** and the resources on the server will not be available.

The Connection details of enterprise server, port number and login name are stored in the user profiles of the system. When you try and login again, these details will be available and you can login by providing you password.

17.4 Accessing the Resources Available on the Enterprise Server

All resources available on the **Enterprise Server** server will be available after the user has been authenticated and has logged into the **Enterprise Server**. Resource in the server has ownership and accessibility criteria associated with it. These resources are arranged and organised into folders and sub folders like any other resource on the system. Further, the owner can associate and manage accessibility of any resource on the enterprise server. The owner can share resources, provide read and write permissions and hide resources from other users.

In addition, resources on the server can be associated with some annotations and meta data. This allows grouping the resources on the server and searching and retrieving the data from the server depending upon the annotations and metadata associated with the resource.

The following functions and features are detailed and discussed in the following sections:

- Browse and manage the resources available on the server
- Open and access files and projects on the server
- Save files and projects on the server
- Upload data, files and projects onto the server
- Change permissions and control accessibility of resources on the server
- Annotate the resources on the server and associate metadata with resources on the server
- Search the retrieve resources from the server based upon the meta data associated with the resource.

17.4.1 Browse and Managing the Resources Available on the Enterprise Server

After a user has logged into the **Enterprise Server** and authenticated by the server, the *Enterprise* tab on the left panel will be populated with all

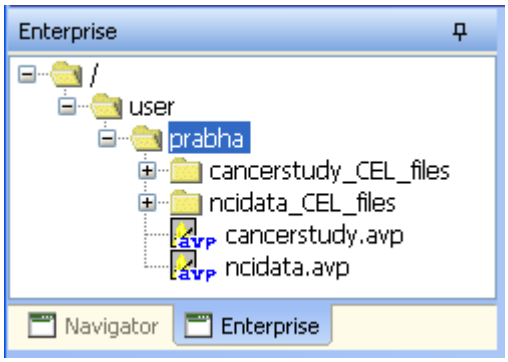


Figure 17.6: The *Enterprise* browser in the left panel

the resource over which the user has appropriate read or write permissions. These will be shown as a tree on the *Enterprise* resource browser on the left panel of the tool.

Navigating the resources in the *Enterprise* is intuitive and like any other resource navigator. The *Enterprise* explorer has many utilities that are available from the Right-Click menu on items in the *Enterprise* Explorer. These are details in the following section on the *Enterprise* Explorer.

17.4.2 Open Projects and Access files from the Enterprise Server

To open and access files form the **Enterprise Server**, use the *Enterprise* $\longrightarrow$ *Open* from the main menu of the tool. This will open a file chooser showing the resources on the **Enterprise Server**. Choose files and click *OK* to load the files in **ArrayAssist**. The file chooser recognizes the files that are relevant for **ArrayAssist**. Thus .avp project files, .CEL and .CHP files can be directly loaded into **ArrayAssist**. Also **ArrayAssist** will identify the type of project (Generic, Affymetrix, Single-dye, or Two-dye projects) and will initiate appropriate action, like loading the corresponding workflow browser, etc.

When project files are opened, the project will be loaded into **ArrayAssist**. The project maintains links to the data files on the **Enterprise Server**. If there are data files associated with the project, like Affymetrix CEL or CHP files or other data files, associated with the project, the user will be prompted with a dialog asking if the data files should be downloaded onto the client. Checking the appropriate check box and clicking *OK* will

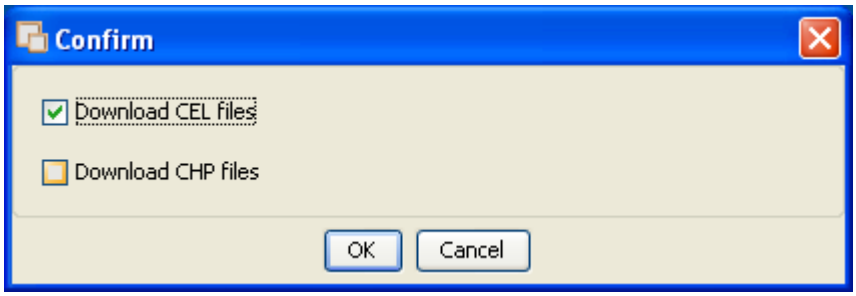


Figure 17.7: Download data files along with the project

download the data files as well on to the client machine. Now the client has all the data and files necessary for the particular project and you will be able to work on the project just like any other project. If the data files are not downloaded onto the client machine, you will not be able to run certain algorithms that may require access to the raw data files, like CEL files.

17.4.3 Creating Projects with data files on the Enterprise Server

The **Enterprise Server** can be used as a data repository with data from microarray experiments loaded on the onto the **Enterprise Server**. The data files may be loaded by administrator of the server or from experimental labs automatically as scheduled tasks. These could be placed onto the **Enterprise Server** into appropriate directories and with appropriate permissions. Setting up such automatic uploads are detailed in the documentation of the **Enterprise Server** and the **Enterprise Manager**.

New projects can be created with data files from the **Enterprise Server**. To create a new project use the *File* $\longrightarrow$*New ... Project* to launch the appropriate project creating wizard. Affymetrix Expression Projects, Affymetrix Exon Projects, Affymetrix Copy Number projects, Singe-dye and Two-dye Projects and the Import Wizard will launch a wizard. In the second step of the wizard, you can choose files from the local file system or from the **Enterprise Server**. To choose files from the **Enterprise Server** you should be logged on to the **Enterprise Server**. If you are logged onto an **Enterprise Server**, on the wizard the *Enterprise...* button will be enabled. Click on the *Enterprise ...* button and will be pop-up a file chooser showing the resources on the **Enterprise Server**. Choose files and create a new project.

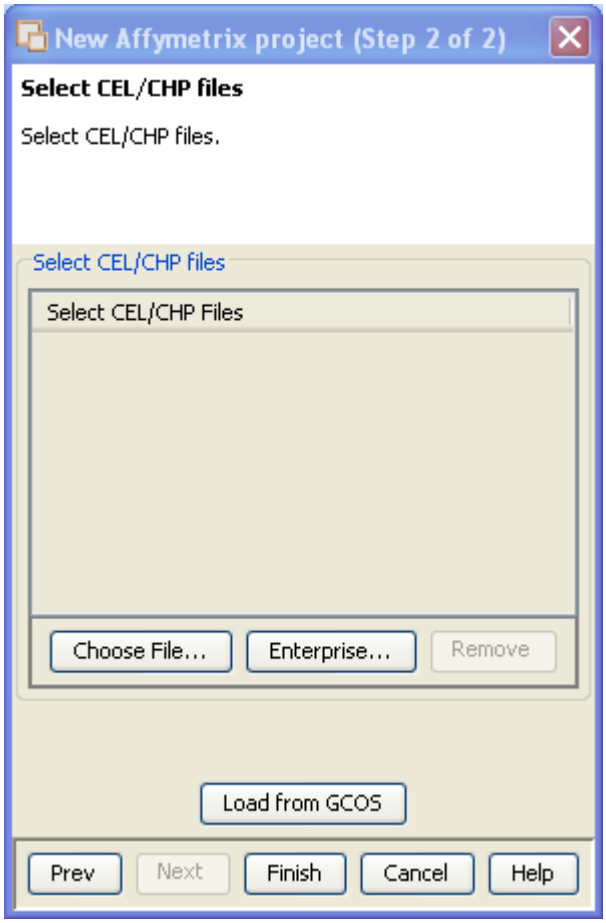


Figure 17.8: Using Data Files for the **Enterprise Server** to Create New Project

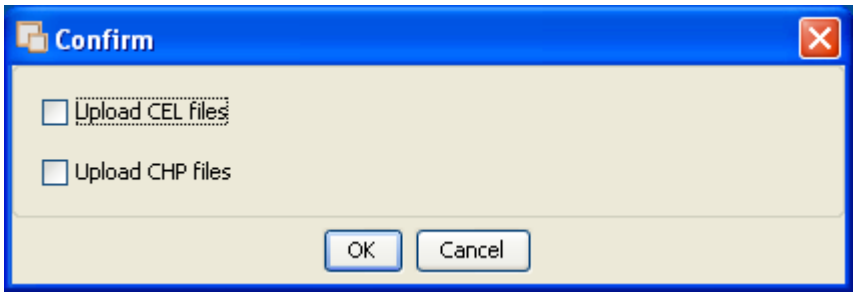


Figure 17.9: Saving project along with data files

17.4.4 Save projects and on the Enterprise Server

You can save projects on the **Enterprise Server** server. These can be accessed over the network and by other clients. If you want to share projects and analysis with other users, you may want to save the project on the server and provide permissions for other users and groups to access the project.

To save projects on the **Enterprise Server**, got to *Enterprise* $\longrightarrow$*Save or Save As* on the main menu bar of **ArrayAssist**. This will pop-up a file chooser showing the directories and files on the **Enterprise Server**. Choose an appropriate folder and click *OK*. This will upload the currently open project on the **Enterprise Server**.

The data files associated with the project are referenced and stored with the project. If a project has been created with data files from the client machine, while saving the project on the **Enterprise Server**, you will be prompted with a dialog asking if the associated data files need to be uploaded and saved along with the project. Clicking *OK* will upload the project along with the data files onto the **Enterprise Server**.

If the project has been created with data files from the **Enterprise Server**, or a project has been opened from the **Enterprise Server** which has data files associated with it, saving the project back to the **Enterprise Server** will automatically only upload the project to the server.

If the project needs to be saved with a different name, click on the *Enterprise* $\longrightarrow$*Save As*. This will open a file chooser dialog showing the directories and files on the **Enterprise Server**. Choose an appropriate folder, provide a name for the project and click *OK*. This will save the project on the **Enterprise Server**.

17.4.5 Loading Data Files and Annotations on the Enterprise Server

Any type of file can be loaded onto the **Enterprise Server** and shared with other users and groups. These features are available from the right-click menu on the *Enterprise* explorer and is detailed in the following section

Annotations has be associated with the files and resources available on the **Enterprise Server**. These annotations are in the form of key - value pairs and is stored as meta data associated with the resources. The client has powerful search and retrieve capability that will search the meta data associated and resource and retrieve resources that satisfy the search criteria. These functions are available on the *Right-click* of the *Enterprise* navigator.

All microarray project can have associated annotations like the experimental grouping information, MIAME annotations, etc. These annotations are associated with the project and its data files. As mentioned earlier, the **Enterprise Server** has an elaborate vocabulary for MIAME annotations. Annotations associated with a project and its data files are automatically saved with the project and uploaded to the server. These annotations can be viewed and searched upon. In addition, the client has the capability to import annotations into a file or multiple files; copy annotations from a file to the clipboard and paste annotations from the clipboard into one or multiple files. These functions are detailed in the following section.

MIAME Annotations for CEL Files

The normal usecase of annotating multiple CEL files uploaded directly onto the **Enterprise Server** server is handled as follows:

- Assume all CEL files are uploaded onto the server using the automatic upload from say a directory on GCOS.
- If the user wants to add miame annotations to all the CEL files, then he needs to do the following:
  - Open the annotation view on one of the CEL files from **ArrayAssist** client.
  - Go through the miame annotations and say *OK*.
  - Then export these annotations to a text file.
  - Then choose all the other CEL files and import the text file with annotations into them. This is done by *Annotation* $\longrightarrow$*Import*

from the right-click on the **Enterprise Server** navigator. Multiple files can be chosen to import annotations on all of them in one go. However, while importing care should be taken that hybridization related information is not imported onto all CEL files. This information is different for each CEL file. To avoid this, either you do not enter hybridization information for the first CEL file itself, or while importing on the other CEL files, choose the rows that do not pertain to hybridization.

- If the user wants to add custom annotations to the CEL files then do the following steps:
  - Create a project using the CEL files.
  - Open miame annotation dialog from within the project.
  - In the custom annotation section, choose import from file option.
  - The file format is simple - its just a tab separated file with three columns the annotation key, value and hybridization name. When this project is saved on the enterprise, the hybridization name from the third column of the custom annotations is used to transfer the annotation information onto all the CEL files.

17.5 The Enterprise Explorer

The enterprise *Explorer* is displayed in the left panel of the tool. When a user connects to the **Enterprise Server**, the explorer shows the resources on the server that are accessible to the user. Resources on which the user has *Read* or *Write* permission will be displayed on the explorer panel as a tree structure.

ArrayAssist supports a whole range of operations on the resources available on the server. These are accessible by selecting a folder or a file on the *Enterprise* explorer and right-click on the selection. This will display a menu of functions that are accessible. The right-click menu on a folder is different from the right-click menu on a file.

17.5.1 Options on Folders on the Explorer

Some of the important functions accessible from the right-click menu are detailed in below:

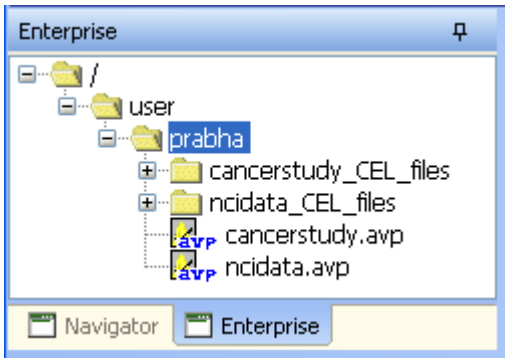


Figure 17.10: Enterprise Explorer

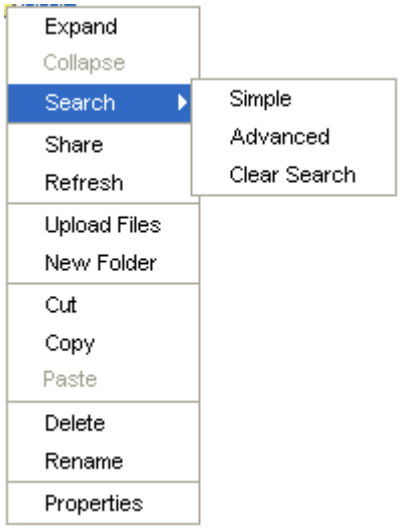


Figure 17.11: Right-click menu on a Folder in the *Enterprise* Explorer

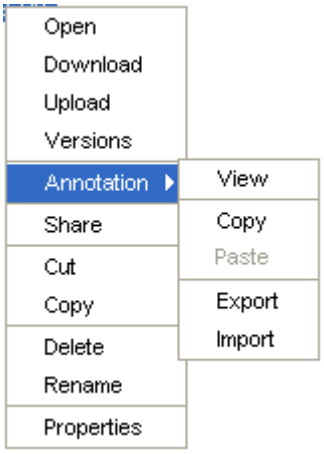


Figure 17.12: Right-click menu on a File in the *Enterprise* Explorer

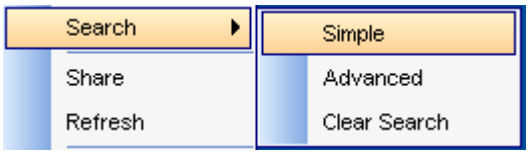


Figure 17.13: The Search menu on Folder Right-Click

Expand and Collapse The folders can be expanded or collapsed by selecting the appropriate option. The appropriate action will be enabled.

Search The search function allows very simple to most complex searches on the resources available on the server. All resources on the server can be annotated with some meta data detailing and describing the resources. These meta data are essentially arranged as a key - value pair. The search function will search the key-value pairs and return the search results in a table table at the bottom of the tool.

- **Simple Search**. Enter key words and this will search all all the annotation values for all resources recursively in the folder. The search results will be displayed in the *Enterprise Search results* in the bottom panel of the tool.
- **Advanced Search**. The Advanced Search feature allows for complex searches on annotations and file attributes. You can

search on file attributes, consisting of the file type or file extension; file name, owner; modified by; file size, creating date and modification date. You can also search by file annotations.

All annotation keys pertaining to the particular file type will be displayed on the *Available Annotations*. You can construct a complex searches from the user interface and combine each search criteria by a OR or AND.

- **Clear Search**. This will clear the *Enterprise Search Results* window.

Share The share utility allows the user to set permissions on a folder. These permissions are applied at the level of the groups an not at the level of individual users. This option will bring up the *Share* dialog where the user can choose a group and provide them *Read* or *Write* permissions. By default directories are created with *No Access* to anyone else except the user.

Refresh This will refresh the *Enterprise* explorer tree and show the current state of the resources on the server.

Upload Files Files from the client machine can be uploaded to the server by choosing the *Upload Files* option. This will pop-up a file chooser. Navigate to the directory and choose the file and click *Open*. Multiple files can be chosen and uploaded together onto the **Enterprise Server**. This will upload all the selected files to the server.

New Folder You may want to create a new folder on the explorer to load files and organize your resources. To do this, select *New Folder*. This will create a new folder on the explorer tree. You can give the folder a name and this will be available.

Cut, Copy, Paste Folders can be cut and placed on the clipboard, copied to the clipboard or pasted from the clipboard into any other location. Once files have been copied to the clipboard, you can *Paste Alias*, where the file is not physically copied, but the copied file is linked from the current location to the original location.

Delete, Rename Folders can be selected and deleted or renamed.

Properties The Folder properties can be viewed and changed from the *Properties* dialog. The owner of the folder, the size and creation and

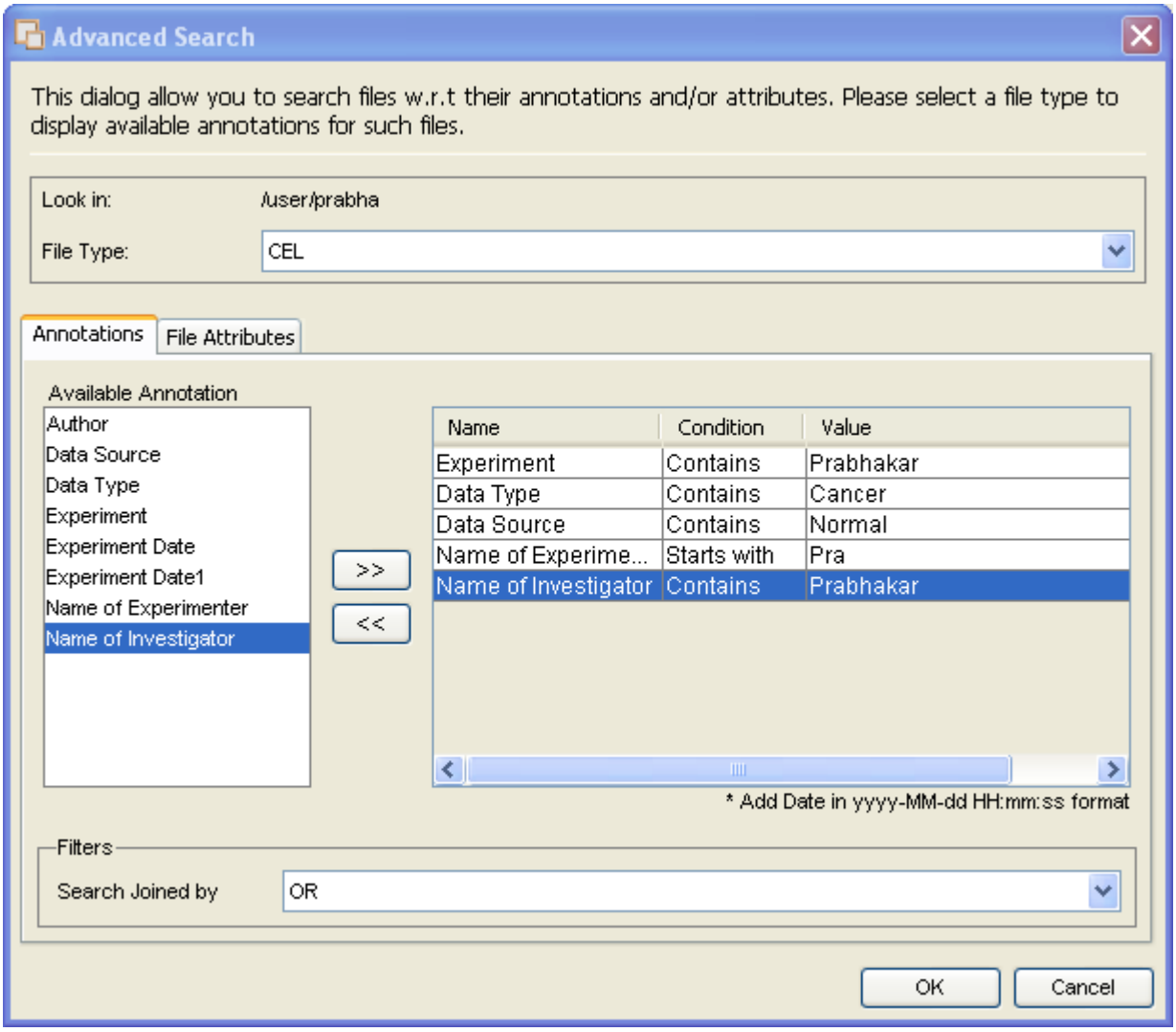


Figure 17.14: Advanced Search Dialog

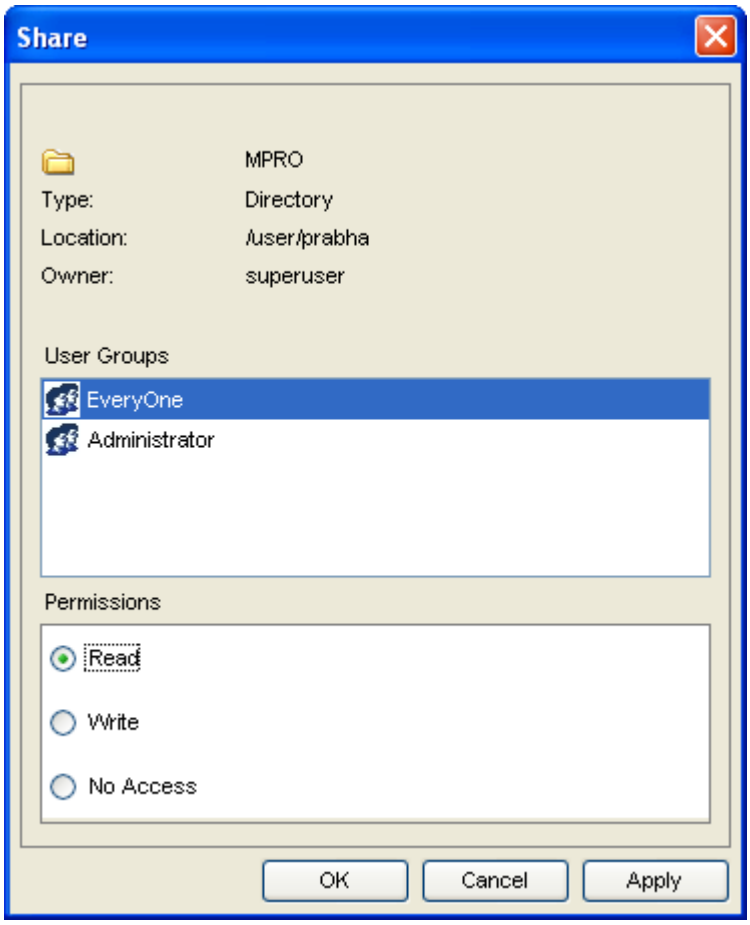


Figure 17.15: Share Dialog on Folders in the *Enterprise* Explorer

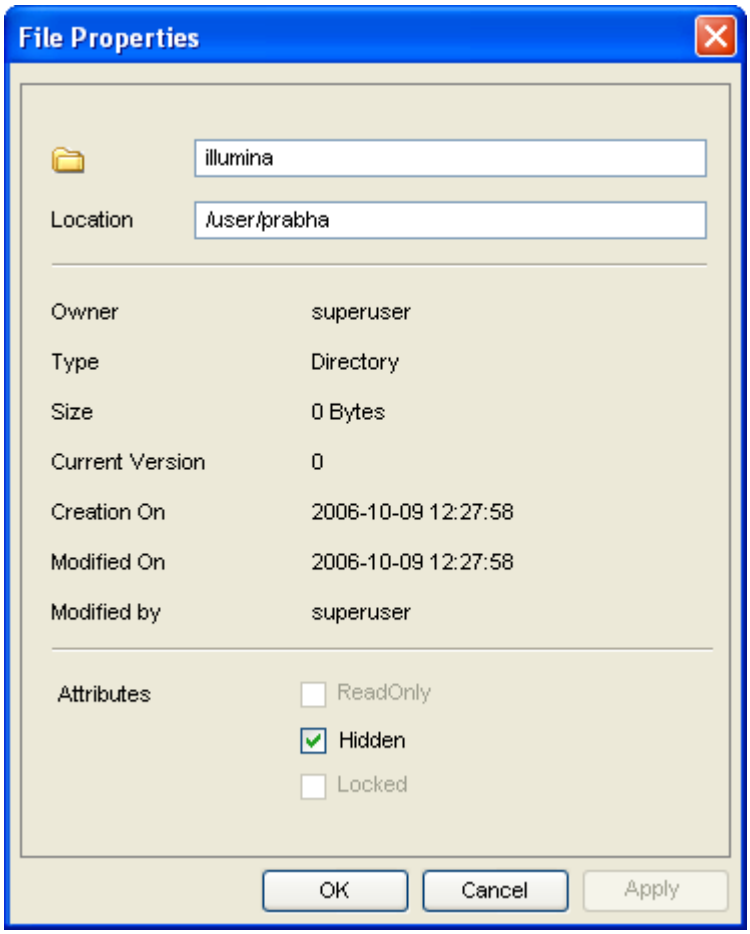


Figure 17.16: Property dialog on Folders in Explorer Tree

modified times can be viewed. Attributes and Folder name can be changed.

17.5.2 Options on Files on the *Enterprise* Explorer

Some of the important functions accessible from the right-click menu are detailed in below:

Open Certain files types like project files can be directly opened in **ArrayAssist**. To open project files, use the *Open* option. This will open the project in **ArrayAssist**. This function is similar to the *Enterprise* $\longrightarrow$*Open*. If data files are associated with the project, while loading the project, you will be asked if the datafiles need to be downloaded onto the client. This function is just like the *Enterprise* $\longrightarrow$*Open* utility.

Download Files from the **Enterprise Server** can be downloaded to the client machine. To download a file from the server use the *Download* function. This will pop-up a file chooser dialog for a location and download the file to the client machine.

Upload You can upload a file from the client machine to **Enterprise Server** by using the *Upload* function. This will pop-up a file chooser. Choose the file and click *Open*. This will upload and replace the particular file on the **Enterprise Server** with the uploaded file.

Versions The **Enterprise Server** has an in-built versioning system that maintains all the previous versions of the file along with the modification date and modified by. Any of the previous modifications can be downloaded and the changes reversed if needed. These versions are maintained along with any annotation associated the resource.

Annotations All files on the **Enterprise Server** can be annotated as key - value pairs and these are stored as meta data for the file. Annotation keys are listed in the *Advanced Search* option and searches can be built to search on the values for each key. Annotations are also specific to the version. If a new version of the file is being uploaded to the **Enterprise Server** then the client application has to attach an appropriate annotation.

- **View** This will show the annotations associated with a file as a table of key-value pairs. **ArrayAssist** shows all the MIAME

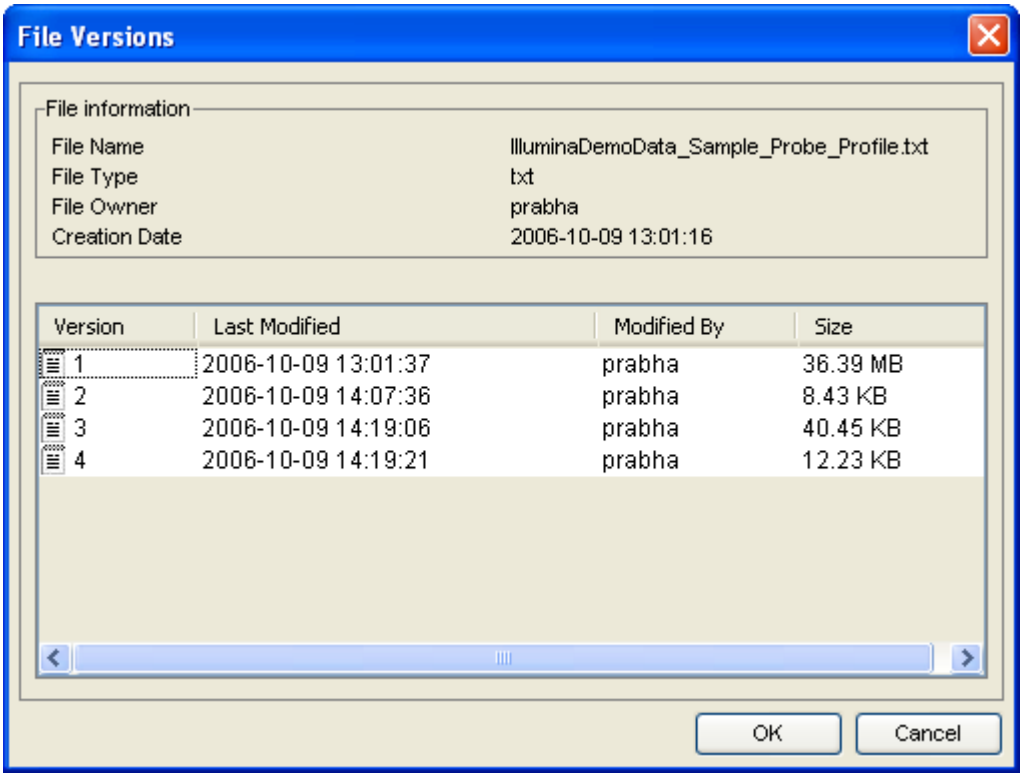


Figure 17.17: File Versions

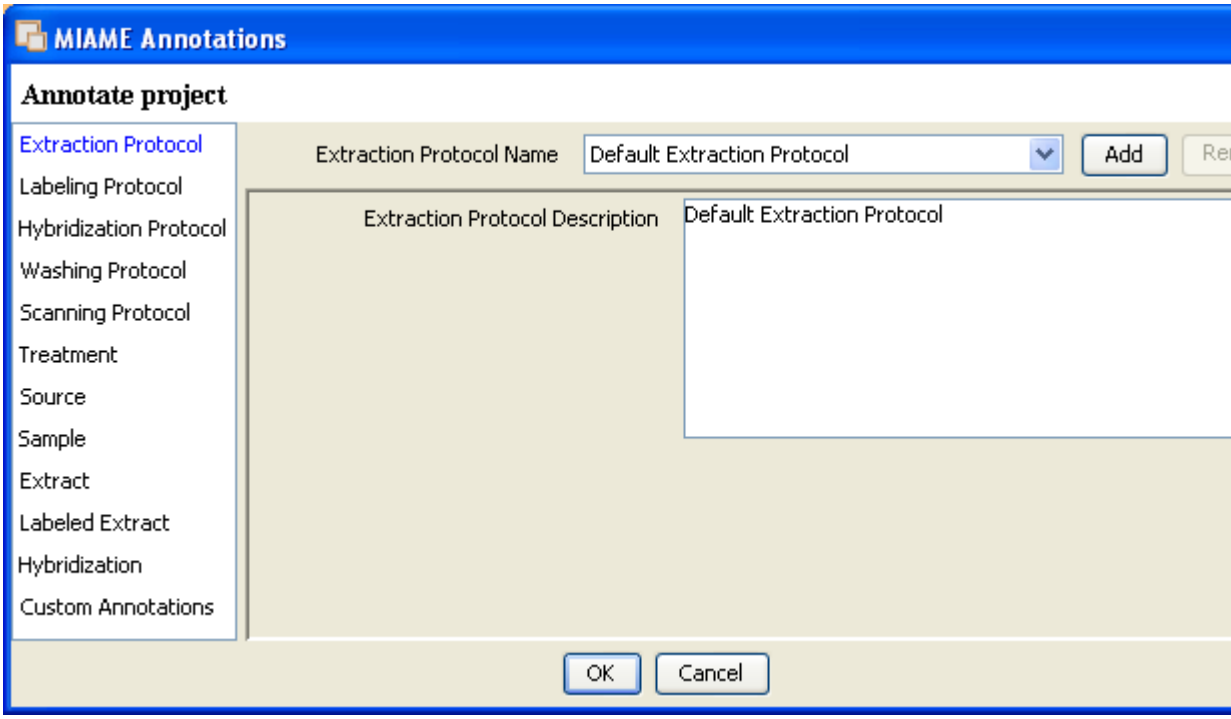


Figure 17.18: Annotation View

annotations as well as the custom annotations added to the files. The screenshot below shows the MIAME annotations.

- **Copy** This copies the annotation for the current file to the clipboard.
- **Paste** This pastes the annotation on the clipboard to selected file or files. Note that annotations can be simultaneously pasted on multiple files.
- **Export** This will export the annotation on the selected file. The *Export Annotation* dialog asks for export details, separator formats and gives a preview and asks for a file name to export.
- **Import** This imports annotation data as a key-value pair from a text file. The format of the annotation can be chosen by a wizard. You can choose different separators and select the columns from a text file that need to be added.

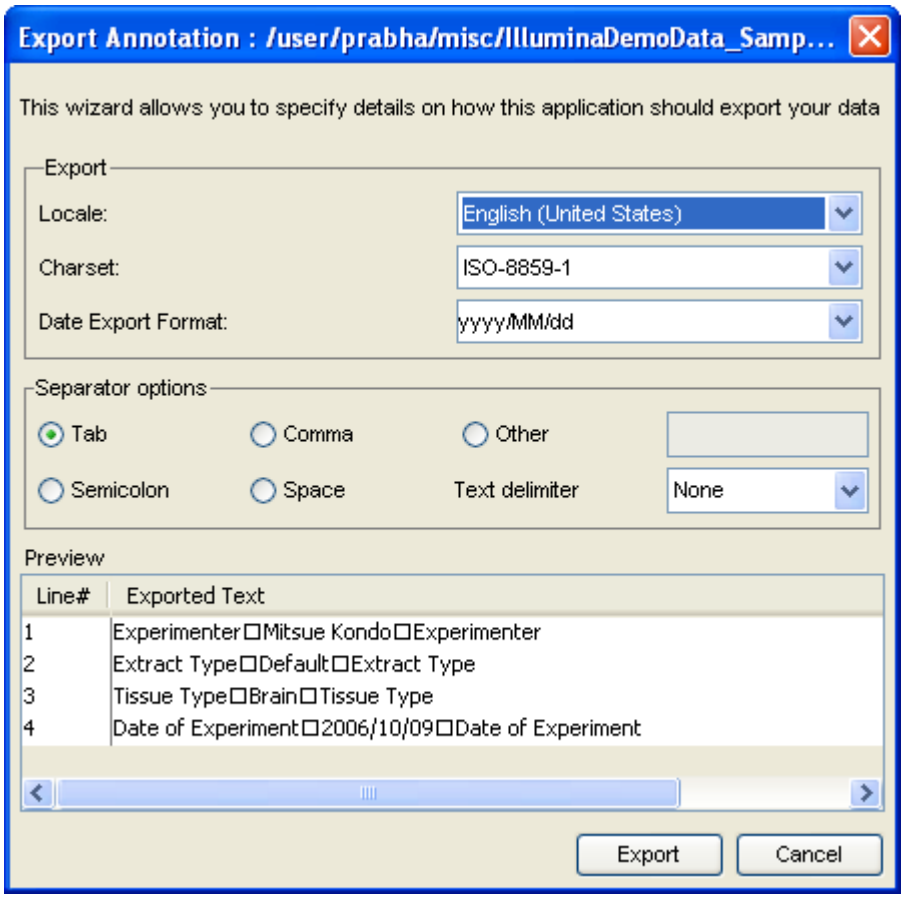


Figure 17.19: Annotation View

Share The share utility allows the user to set permissions on individual files. These permissions are applied at the level of groups an not at the level of individual users. This option will bring up the *Share* dialog where the user can choose a group and provide em Read or *Write* permissions. By default files are created with *No Access* to anyone else except the user.

Cut, Copy, Paste Files can be cut and placed on the clipboard, copied to the clipboard or pasted from the clipboard into any other location. Once files have been copied to the clipboard, you can *Paste Alias*, where the file is not physically copied, but the copied file is linked from the current location to the original location.

Delete, Rename Files can be selected and deleted or renamed.

Properties The File properties can be viewed and changed from the *Properties* dialog. The owner of the folder, the size and creation and modified times can be viewed. Attributes and Folder name can be changed.

17.6 Migrating data from the Gene Traffic Enterprise Server

NOTE: You will need to have administrative privileges for migrating Gene Traffic projects to the **Enterprise Server**.

An Enterprise server 1.x is being launched that will replace the current Gene Traffic server and provide an integrated and scalable solution to the analysis of the whole of microarray data.

The **ArrayAssist** along with the **Enterprise Server** is the next generation **Enterprise Server** from Stratagene's **Gene Traffic Server**. All Gene Traffic Affymetrix and Two-Dye projects will be automatically migrated to an ArrayAssist project and uploaded to the Enterprise Server with the **ArrayAssist** enterprise client module. Note that you will need administrative privileges on the Gene Traffic Server and the **Enterprise Server** to do the migration. The server administrator will normally be the person who would do the migration.

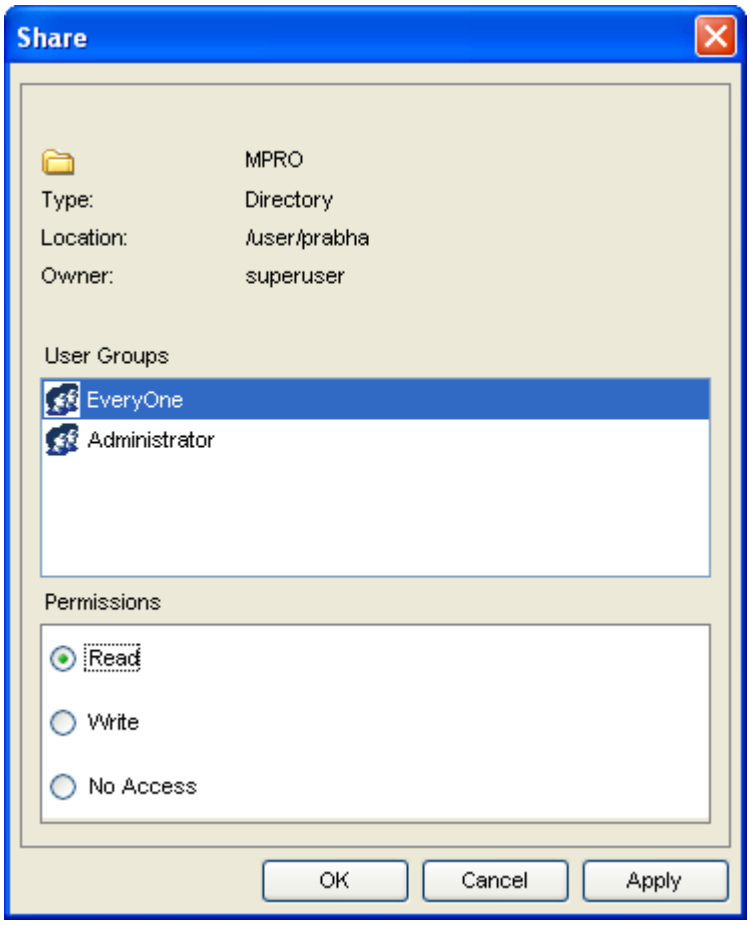


Figure 17.20: Share Dialog on Files in the Explorer

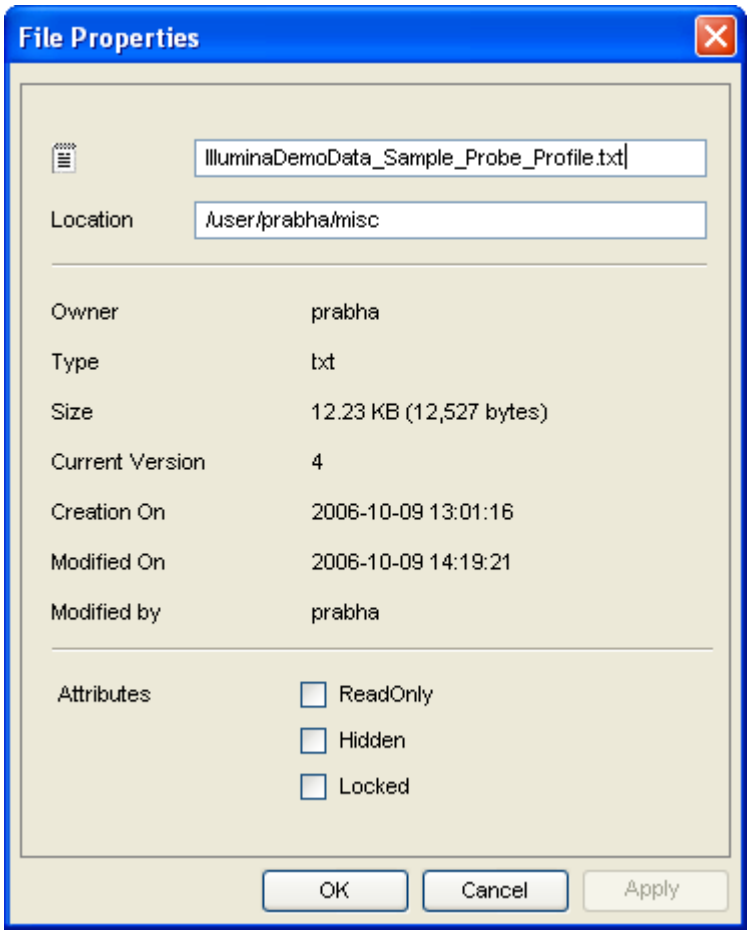


Figure 17.21: Property dialog on Files in Explorer Tree

17.6.1 Requirements

- You should have Gene Traffic 3.2-11. If you do not have 3.2-11, you will have to upgrade to this version from the web.
- You should have Array Assist **Enterprise Server** version 1.0 installed and running. You should have created a directory with enough disk space for the AA Enterprise user repositories. This directory may be called D`EnterpriseData`
- You should have ArrayAssist Client 5.0.x installed and activated on any machine on the network. You should be able to access the Gene Traffic server as well as the ArrayAssist **Enterprise Server** from the ArrayAssist Client.
- You should have the script DBpasswords.sh. This is used to reset and restore the password for all users on the GT server. This script must be placed on the GT server.

17.6.2 Preparing for Migration on GT server

- Make sure no users and logged onto the GT server.
- Reset the username and password for all users on the GT server:
  - Copy the script DBpasswords.sh to GT server.
  - Log on to GT Server in a secure shell as root. Execute the script by issuing the following command

    `./DBpasswords.sh --reset`
  - This will prompt for the password for the user apache of the database on the GT Server. This is usually a blank.
  - After authenticating the password, it will run the script and set all user passwords to default (**except the password of the admin**)
  - This will also create a file called Passwords.csv in the same folder from which the script was run.
  - Copy this file (Passwords.csv) to the machine with AA5.0 client. The password file will be necessary when you run the migration script and get projects from the GT Server.
  - After the migration process is fully complete, you can restore old passwords by running the script as

    ./DBpasswords.sh –restore

- This will restore the original passwords for all users on the GT server. Users will be able to login again into the GT server.

- The project summary for all projects on the GT server will need to be cleaned up by issuing the following commands on the GT server as root.

```
cd /var/www/html/projects for file in ‘ls‘; do mkdir
cp $file/data.SAV mv $file/data/Project.zip
$file/data.SAV done
```

17.6.3 Preparation for Migration on ArrayAssist machine

- You should have ArrayAssist5.0 client version installed and activated. You should be able to connect to the GT server as well as the enterprise server from the Client machine. You will need to have enough disk space on the Client machine since all chosen GT projects will be downloaded onto the client.

- Library files for all organisms for which projects exist on the GT server should be available apriori on the client from which the migration is being trigerred. Go to *Tools* $\longrightarrow$*Update Data Library* $\longrightarrow$*From Web* and click on *Show Available Updates* button in the dialog that comes up. From the list of updates, choose the GeneChip libraries for which there are projects on your GT server or just update the entire pack of library files. The whole pack will take about 1.5GB of disk space.

- Create two directories on the client machine where temporary files and intermediate project files will be stored. For example, on windows:

```
C:/Migration/TMP (to store all the temporary file)
C:/Migration/DATA (to create and keep AA project files)
Copy the Passwords.csv file to C:\Migration\TMP.
Also make sure that
C:\Migration\DATA is empty.
```

- If you are connected to the enterprise server, then disconnect using *Enterprise* $\longrightarrow$*Disconnect* menu.

17.6.4 Running the Migration

- Open any avp file and open the script editor and type and run the following command.

  `script.enterprise.gtmigration.start()`

- This will show an information dialog. Please read this carefully before proceeding with the migration.

- This will popup a dialog where you will have to enter the following details: A screenshot of the dialog is shown below.

  - Gene Traffic Server Details:
    * Host IP: Enter the host IP address of the GT Server
    * Login: Enter the admin user
    * Password: Enter the admin user password
  - Download Folders:
    * For temporary files: Enter the directory For temporary files on the AA Client. This directory should contain the Passwords.csv file For Example on windows `C:\Migration\TMP`
    * For Project data: Enter the directory where intermediate project files are stored on the AA Client. This directory should be empty. For example: `C:\Migration\DATA`
  - **Enterprise Server** Details:
    * Host IP: Enter the host IP for AA Enterprise
    * Post: Enter the Port on AA Enterprise (8080)
    * Login: Enter superuser
    * Password: Enter the password for the superuser. The default password is strand123

- This will login to the GT Server and the AA **Enterprise Server** with the login details provided and popup a dialog for the location of the Repository Root on the AA **Enterprise Server**. Click on the dropdown arrow. This will open file chooser with the file system of the AA **Enterprise Server**. Here choose a directory as a Repository where repositories will be created for each user and the users files will be migrated into the repository.

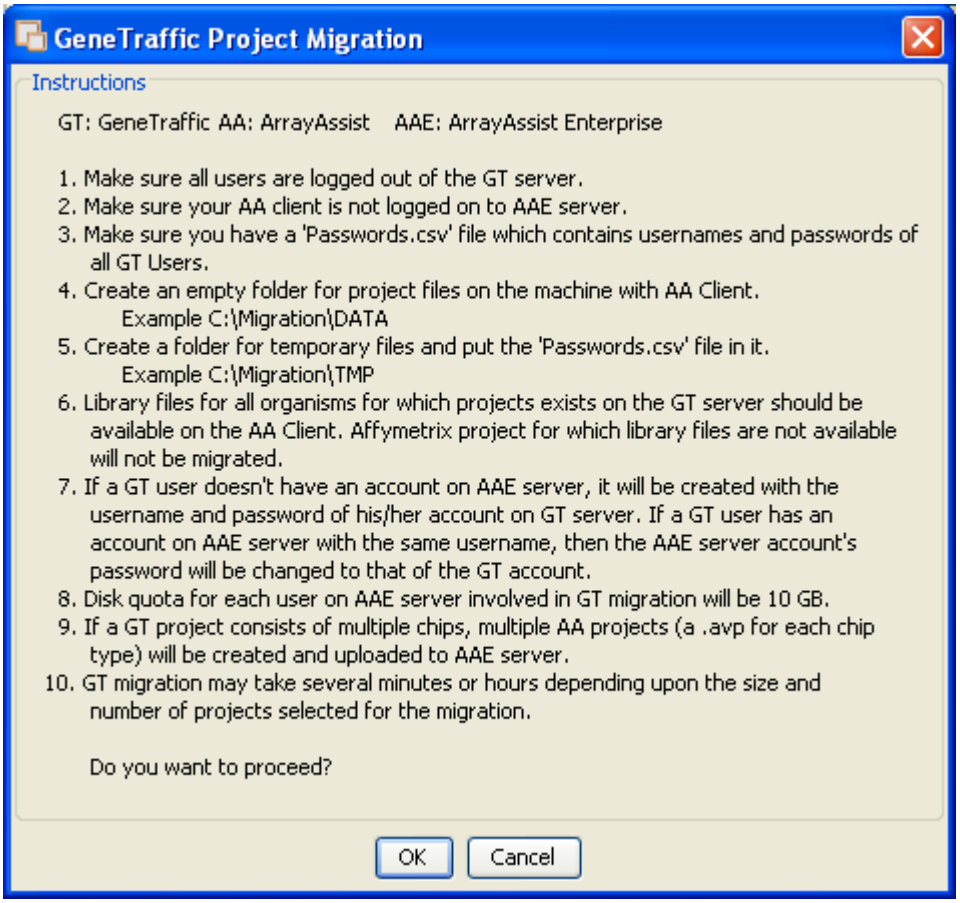

GeneTraffic Project Migration
Instructions
GT: GeneTraffic AA: ArrayAssist AAE: ArrayAssist Enterprise
1. Make sure all users are logged out of the GT server.
2. Make sure your AA client is not logged on to AAE server.
3. Make sure you have a 'Passwords.csv' file which contains usernames and passwords of all GT Users.
4. Create an empty folder for project files on the machine with AA Client.
Example C:\Migration\DATA
5. Create a folder for temporary files and put the 'Passwords.csv' file in it.
Example C:\Migration\TMP
6. Library files for all organisms for which projects exists on the GT server should be available on the AA Client. Affymetrix project for which library files are not available will not be migrated.
7. If a GT user doesn't have an account on AAE server, it will be created with the username and password of his/her account on GT server. If a GT user has an account on AAE server with the same username, then the AAE server account's password will be changed to that of the GT account.
8. Disk quota for each user on AAE server involved in GT migration will be 10 GB.
9. If a GT project consists of multiple chips, multiple AA projects (a .avp for each chip type) will be created and uploaded to AAE server.
10. GT migration may take several minutes or hours depending upon the size and number of projects selected for the migration.
Do you want to proceed?
OK
Cancel

Figure 17.22: Gene Traffic Migration Intsructions Dialog

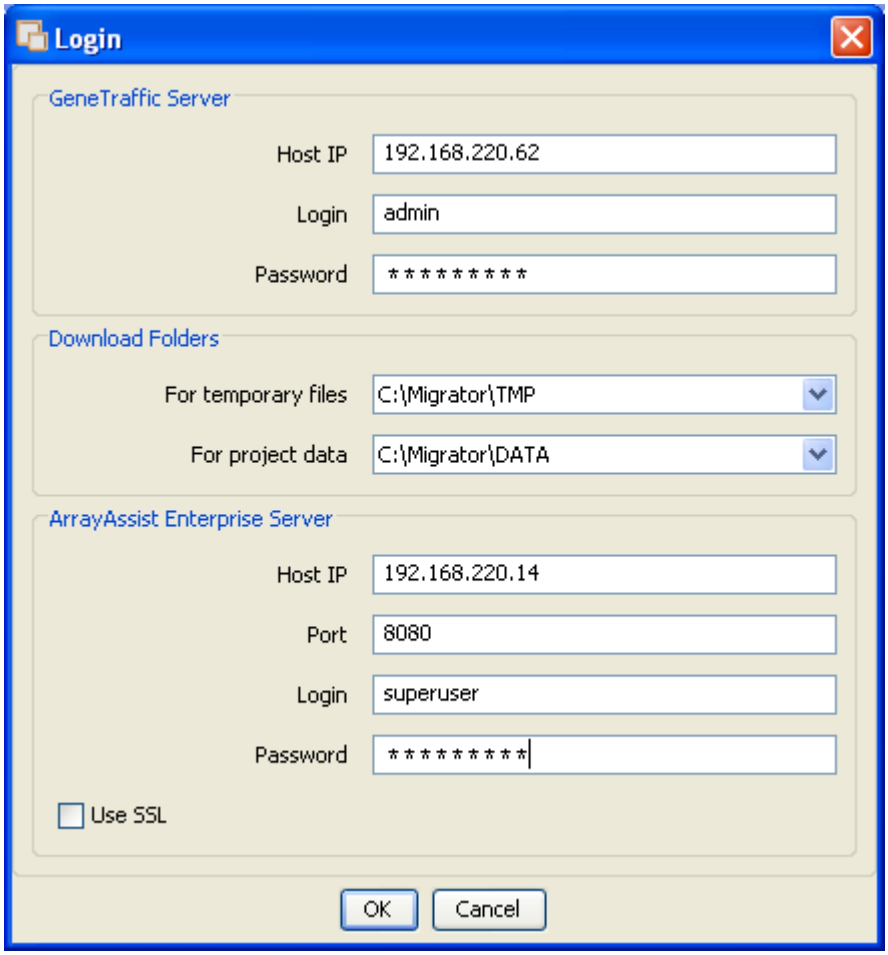


Figure 17.23: Gene Traffic Migration Login Dialog

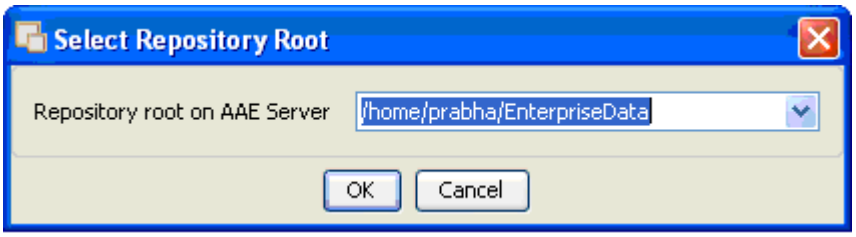


Figure 17.24: Choose Root Repository on **Enterprise Server**

It is a good practice to have a repository folder called EnterpriseData within which all user repositories will be created.

By default, each user will be created with a disk quota of 10 GB. If a user has projects more than 10 GB, the migration of projects in excess of 10 GB limit will fail, and will be shown as failed in the Report. Migrating the remaining projects will require manual intervention.

- The next screen shows the list of users on the GT Server and the Affymetrix and Two-Dye projects for each user. Select the users and the projects required to be migrated to the AA Enterprise server abd click OK. Only the selected users and projects will be migrated.
- Now the script will run through the following steps:
  - Step 1: The script will extract the projects from the GT server and create AA Projects for each of them.
    - The script will create a project summary for each project on the GT server.
    - The script will then transfer the project summary files and data file for each project onto the AA 5.0 Client machine and place then in appropriate directories. (make sure you have enough space on the AA 5.0 Client machine to store all the project summary and data files. This may be huge).
    - Note that this process may also take time.
  - Step 2: Create .avp project files for all GT projects. In this process, the AA project will be created with the following information from the corresponding GT project:
    - CEL/CHP files with which the original GT project was created.

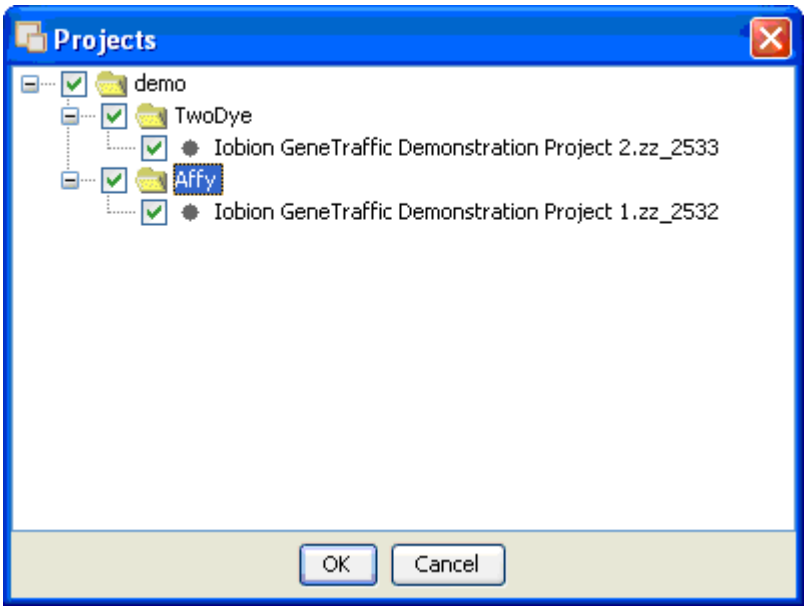


Figure 17.25: Choose Projects for Migration

  - MIAME annotation.
  - Experiment grouping information.
  - A summarized dataset with the name *Legacy GeneTraffic Summarized Dataset*.
  - All data files from the Data Manager part of the GT project will be exported as is and will not be imported into the AA project. These will be uploaded onto the AAE server in the same place as the AA project, and can be imported into the project by the user at a later stage if required.
- Step 3: Create accounts for each GT user in the AA Enterprise and allocate a repository for the user under the chosen Repository Root and load the projects created on the AA Client onto the AA Enterprise server.
  - It will create all the GT accounts with the password `default` on the AAE Server.
  - For each user, it will then login as that user and upload the users .avp projects, CEL/CHP files and data files to the AAE server, with appropriate user permissions.

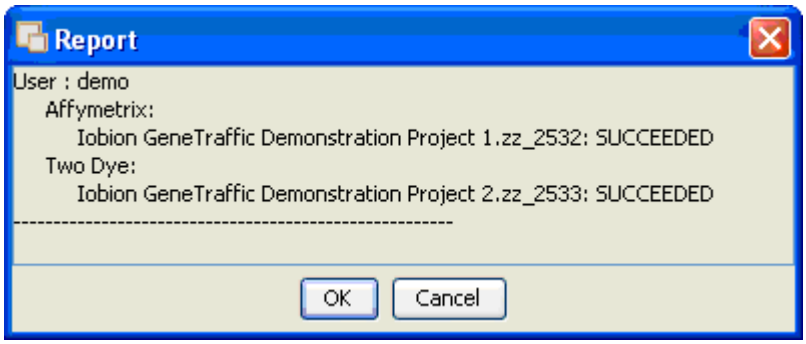


Figure 17.26: Gene Traffic Migration Report

- The migration process may take several minutes to many hours depending upon the number of projects selected for migration.
- After the migration is complete a report is presented stating the number of projects migrated with failures or errors in any.
- Note that ArrayAssist does not support a project with multiple chip types while GeneTraffic supports such projects. In GT projects with multiple chip types, two corresponding projects will be created in ArrayAssist.

17.6.5 Post-Migration Cleanups and Restore:

- After the migration is complete, review the report to see if all the projects have been migrated. You can save the report to a file when you dispose the Report dialog.
- Finally you will restore all the user passwords on the GT server to the original. To do this, login to your GT server as root, and run the following command. ./DBpasswords.sh –restore This command will restore all the passwords of all users to the original GT password.
- The GT server and the AA Enterprise server can now be open to users.

Chapter 18

Scripting

18.1 Introduction

ArrayAssist offers full scripting utility which allows operations and commands in **ArrayAssist** to be combined within a more general Python programming framework to yield automated scripts. Using these scripts, one can run transformation operations on data, automatically pull up views of data, and even run algorithms repeatedly, each time with slightly different parameters. For example, one can run a Neural Network repeatedly with different architectures until the accuracy reaches a certain desired threshold.

To run a script, go to Tools→Script Editor. This opens up the following window. Write your script into this window and click on Run  icon to execute the script. Errors, if any, in the execution of this script will be recorded in the Log window. You can also stop script execution at user defined breakpoints by pressing Stop  icon. For convenience in debugging, clicking on a row in the script editor highlights the row number in the ticker at the bottom.

This chapter provides a few example scripts to get you started with the powerful scripting utility available in **ArrayAssist**. An ehaustive and extensive scripting documentation to exposes all functions of the product is in preperation and will be released shortly. Utility and example scripts from the development team as well as from **ArrayAssist** users will be constantly updated at the product website.

The example scripts are divided into 4 parts: Dataset Access, Views, Commands and Algorithms, each part detailing the relevant functions available. Note that to use these functions in a Python program, you will need some knowledge of the Python programming language. See http://www.

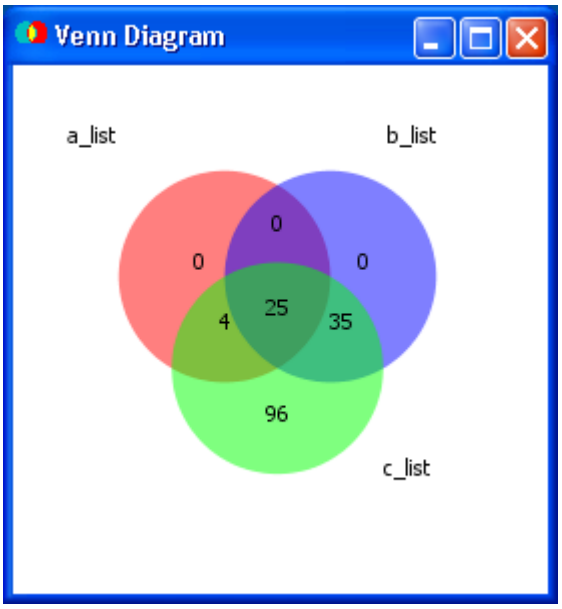


Figure 18.1: Scripting Window

`python.org/doc/tut/tut.html` for a Python tutorial. Example scripts in the samples folder of the **ArrayAssist** install directory can also serve as good starting points to learn scripting. Please note that tabs and spaces are important in python and denote a block of code.

Note: The scripts provided here can be pasted into the Script Editor and run.

18.2 Scripts to Access projects and the Active Datasets ArrayAssist

18.2.1 List of Project Commands Available in ArrayAssist

```
###################### PROJECT OPERATIONS
#
#
## commands and operations
#
#
```

```
########################################
#
## Imports the package required for project calls
#

from script.project import *

########## getProjectCount
#
## This return the number of projects that are open.
#

a = getProjectCount()
print a

########## getProject(index)
#
## This returns a project with the that index from [0,1...]
#

a = getProject(0)
print a.getName()

########## getActiveProject():w
#
## This return the active project.
#

b = getActiveProject()
print b

########## setActiveProject(project)
#
## This sets the active project to the one specified.
## The active project must be got with the getProject() command
## The project here is got by a = getProject(0)
#

setActiveProject(a)
```

```
########## removeProject(project)
#
## This removes the project from the tool.
#

removeProject(getProject(1))


########## ACCESSING ELEMENTS IN PROJECT ############
#
#
## commands and operations
#
#
##########################################

########## getActiveDatasetNode()
#
#This returns the active dataset node from the current project
#
a = getActiveDatasetNode()
print a

## getActiveDataset()
#
# This return the active dataset on which operations can be performed.
#

a = getActiveDataset()
print a

########## getFocussedViewNode()
#
## This return node of the current focussed view.
#

a = getFocussedViewNode()
print a

########## getFocussedView()‘
```

```
#
## This gets the current focussed view on which operations can performed
#

a = getFocussedView()
print a

#
##    class PyProject: the methods defined here in this class
##     work on an instance of PyProject which can be got using the
##     getActiveProject() method defined in script.project
#

########## getName()
#
## This returns the name of the current active project
#

p = getActiveProject()
print p.getName()

########## setName(name)
#
## This will set a name for the active project
##

p.setName('test')

########## getRootNode()
#
## This will return the root node (master dataset) on which
## operations can be performed.

rootnode = p.getRootNode()
print rootnode.name

########## getFocussedViewNode()
#
## This will return the node of the current focussed view on
## which operations can be performed
```

```
#

f = p.getFocussedViewNode()
print f.name

########## setFocussedViewNode(node)
#
## This gets a view with the given title and brings its node
## in focus.
#

v = script.view.getViewWithTitle("Scatter Plot")
s = p.setFocussedViewNode(v.getNode())


########## getActiveDatasetNode()
#
## This returns the current active dataset node in the project
#

d = p.getActiveDatasetNode()
print d.name

########## setActiveDatasetNode(node)
#
## This will take in a dataset node and set that as active
#

p.setActiveDatasetNode(p.getRootNode())


#
##    class PyNode: the methods defined here in this class
##     work on an instance of PyNode which can be got using the
##     get*****Node() methods defined in class PyProject
#

########## getName()
#
## This will return the name of the node with which it is called
```

```
#

node = p.getFocussedViewNode()
print node.getName()

########## getDataset()
#
## This returns the dataset fro the dataset node with which it is
## called.
#

node = p.getRootNode()
dataset = node.getDataset()
print dataset.getName()

########## getChildCount()
#
## This returns the number of children of the node with which
## it is called.
#

count = node.getChildCount()
print count

########## getChildNode(key)
#
## This returns the child node having name equal to key.
#

child = node.getChildNode("LR Train")
print child.getName()

########## addChildFolderNode(node)
#
## This will add a chile folder node with the name specified.
#


########## addChildDatasetNode(name, rowIndices=None, columnIndices=None, setActive=1, ad
#
```

```
## This will create a subset dataset, with the given row and
## column indicies and add it as a child node.
#

node.addChildDatasetNode("subset", rowIndices=[1,2,3,4,5], columnIndices=[0,1], s
```

18.2.2 List of Dataset Commands Available in ArrayAssist

```
####################### DATASET OPERATIONS
#
#
## commands and operations
#
#
########################################

from script.dataset import *

##########    - parseDataset(file)
#
## This allows creating a dataset by parsing the given file
#

##########    - writeDataset(dataset, file)
#
## This allows to save a given dataset to a file
#

##########    - createIntColumn(name, data)
#
## This allows to create a Integer column with the specified name
## having the given data as values
#

##########    - createFloatColumn(name, data)
#
## This allows to create a Float column with the specified name
```

```
## having the given data as values

##########    - createStringColumn(name, data)
#
## This allows to create a String column with the specified name
## having the given data as values
#
#

#
#
##    class PyDataset: The methods defined here in this class
##     work on an instance of PyDataset which can be got using the
##     getActiveDataset() method defined in script.project
#

########## getRowCount()
#
## This returns the row count of the dataset
#

dataset = script.project.getActiveDataset()
rowcount = dataset.getRowCount()
print rowcount


########## - getColumnCount()
#
## This returns the column count of the dataset
#

colcount = dataset.getColumnCount()
print colcount


########## - getName()
#
## This returns the name of the dataset
#
```

```
name = dataset.getName()
print name

########## - index(column)
#
## This returns the index of the specified column
#

col = dataset.getColumn('flower')
idx = dataset.index(col)
print idx

########## - __len__(): returns column count
#
## This method is similar to the getColumnCount() method
#

########## - iteration c in dataset:
#
## This iterates over all the columns in the dataset.
#

for c in dataset:
   name = c.getName()
   print name

########## - d[index]
#
## This can be used to access the column occuring at the
## specified index in the dataset.
#

col = dataset[0]
print col.getName()


##########    - getContinousColumns()
#
## This returns all countinuous columns in the dataset.
#
```

```
z = dataset.getContinuousColumns()
print z

##########    - getCategoricalColumns()
#
## This returns all categorical Columns in the dataset.
#

z = dataset.getCategoricalColumns()
print z

##########    class PyColumn: The methods defined in this class
## work on an instance of PyColumn which can be got
## using the getColumn(name), getColumn(index) methods
## defined in the class PyDataset
#
##
#

########## - getSize()
#
## This returns the size of the column which is the same as the
## row count of the dataset.
#

col = dataset.getColumn(0)
size = col.getSize()
print size

########## - __len__()
#
## This is the same as the getSize() method
#

########## - getName()
#
## This returns the name of the column
#
```

```
name = col.getName()
print name


########## - setName(name)
#
## This sets the name of the column to the specified value
#

col.setName('test0')
print col.getName()


########## - iteration for x in c:
#
## This iterates over all the elements in the column
#

for x in col:
    print x

########## - access c[rowindex]
#
## This can be used to access the element occuring at the
## specified row index in the column.
#

value = col[0]
print value



########## - operations +, -, *, /, **, log, exp
#
## This allows mathematical operations on each element in the column
#

d = dataset[1] + dataset[2]
print d[0]
```

18.2.3 Example Scripts

The first example below show how to select rows from the dataset based on values on a column. The second example shows how to append a column to the dataset based on some arithmetic operations and then launch views with those columns.

```
#********************Example*****************************
#
# create a subset with rows where the first column has value 'Iris-setosa'
#

node = script.getActiveDatasetNode()
d = node.getDataset()

def findMatchingIndices(c, name):
    "Returns indices of rows, whose value in the specified column is name"
    return [i for i in xrange(c.getSize()) if c[i] == name]

name = "Iris-setosa"
rowIdices = findMatchingIndices(d[0], name)
colIndices = [0, 1, 3]


node.addChildDatasetNode(name, rowIdices, colIndices)

script.view.Table().show()

#********************Example*****************************
#
# script to append columns using arithemetic operations on columns
#

from script.view import ScatterPlot
```

```
from script.omega import createComponent, showDialog

d = script.project.getActiveDataset()

#
#  define a function for opening a dialog
#


def openDialog():
    A = createComponent(type='column', id='column A', dataset=d)
    B = createComponent(type='column', id='column B', dataset=d)
    C = createComponent(type='column', id='color by', dataset=d)

    g = createComponent(type='group', id='MVA Plot', components=[A, B, C])

    result = showDialog(g)

    if result:
return result['column A'], result['column B'], result['column C']
    else:
return None

#
# define a function to show the plot with two columns of the
# active dataset and show the results
#


def showPlot(avg, diff, color):

    plot = script.view.ScatterPlot(title = 'MVA Plot', xaxis=avg, yaxis=diff)
    plot.colorBy.columnIndex = color

    plot.show()

#
# main
# This will open a dialog, and take inputs
# Compute the average and difference
```

```
# Appened the columns to the dataset
# Show the Plot
#

result = openDialog()


if result:
    a, b, col = result
    avg = (d[a] + d[b])/2
    diff = d[a] - d[b]

    avg.setName('average')
    diff.setName('difference')

    d.addColumn(avg)
    d.addColumn(diff)

    x = d.indexOf(avg)
    y = d.indexOf(diff)
    color = d.indexOf(col)


    showPlot(x, y, color)
```

18.3 Scripts for Launching View in ArrayAssist

18.3.1 List of View Commands Available Through Scripts

The scripts below show how to launch any of the data views and how to close the view through a script.

```
###############Spreadsheet###############
# View : Table
# Creating...
view = script.view.Table()
# Launching...
view.show()
```

```
# Closing...
view.close()

#############Scatter plot##################

# View : ScatterPlot
# Creating...
view = script.view.ScatterPlot()
# Launching...
view.show()
# Changing parameters
view.colorBy.columnIndex=-1
# Closing...
view.close()

#############Heat Map######################

# View : HeatMap
# Creating...
view = script.view.HeatMap()
# Launching...
view.show()
# Closing...
view.close()

#############Histogram#####################

# View : Histogram
# Creating Histogram with parameters...
view = script.view.Histogram(title="Title", description="Description")
# Launching...
view.show()
# Closing...
#view.close()

#############Bar Chart#####################

# View : BarChart
# Creating...
view = script.view.BarChart()
```

```
# Launching...
view.show()
# Closing...
view.close()

##############Matrix Plot#########################

# View : MatrixPlot
# Creating...
view = script.view.MatrixPlot()
# Launching...
view.show()
# Closing...
view.close()

##############Profile Plot#########################

# View : ProfilePlot
# Creating...
view = script.view.ProfilePlot()
# Launching...
view.show()
# Setting parameters
view.displayReferenceProfile=0
# Closing...
#view.close()

#############
```

18.3.2 Examples of Launching Views

The Example scripts below will launch a view with some parameters set.

```
#********************Example*****************************
#
# views that work on individual columns
#
#
```

```
from script.view import *

from script.framework.data import createIntArray


# open ScatterPlot
ScatterPlot(xaxis=1, yaxis=2).show()

# open histogram on column#2
Histogram(column = 2).show()

#********************Example*****************************
#
# views that work on multiple columns
#

indices = [1, 2, 3]

# open box-whisker
BoxWhisker(columnIndices=indices).show()

# open MatrixPlot
MatrixPlot(columnIndices = indices).show()

# open Table
Table(columnIndices=indices).show()

# open BarChart
BarChart(columnIndices=indices).show()

# open HeatMap
HeatMap(columnIndices = indices).show()

# open ProfilePlot
ProfilePlot(columnIndices = indices).show()

# open SummaryStatistics
SummaryStatistics(columnIndices=indices).show()
```

```
#********************Example*****************************
#
# script to open scatterplot with desired properties
#

# import all views
from script.view import ScatterPlot

from script.omega import createComponent, showDialog

dataset = script.project.getActiveDataset()

def openDialog():
    x = createComponent(type='column', id='xaxis', dataset=dataset)
    y = createComponent(type='column', id='yaxis', dataset=dataset)
    c = createComponent(type='column', id='Color Column', dataset=dataset)

    g = createComponent(type='group', id='ScatterPlot', components=[x, y, c])

    result = showDialog(g)

    if result:
return result['xaxis'], result['yaxis'], result['Color Column']
    else:
return None

def showPlot(x, y, c):

    plot = script.view.ScatterPlot(xaxis=x, yaxis=y)
    plot.colorBy.columnIndex = c

    # set minColor to red. just giving RGB components is enough
    plot.colorBy.minColor = 200, 0, 0

    # set maxColor to blue
    plot.colorBy.maxColor = 0, 0, 200
    plot.show()
```

```
result = openDialog()

if result:
    x, y, c = result
    showPlot(x, y, c)
```

18.4 Scripts for Commands and Algorithms in ArrayAssist

18.4.1 List of Algorithms and Commands Available Through Scripts

```
############

# Algorithm : log
# Parameters: base, outputOption, prefix, childDatasetName,
# Creating...
algo = script.algorithm.log()
# Executing...
algo.execute(displayResult=1)

#############

# Algorithm : exponent
# Parameters: base, outputOption, prefix, childDatasetName,
# Creating...
algo = script.algorithm.exponent()
# Executing...
algo.execute(displayResult=1)

#############

# Algorithm : absolute
# Parameters: outputOption, prefix, childDatasetName,
# Creating...
algo = script.algorithm.absolute()
# Executing...
algo.execute(displayResult=1)
```

```
##############

# Algorithm : scale
# Parameters: scaleFactor, scaleType, outputOption, prefix, childDatasetName,
# Creating...
algo = script.algorithm.scale()
# Executing...
algo.execute(displayResult=1)

##############

# Algorithm : threshold
# Parameters: min, max, outputOption, prefix, childDatasetName,
# Creating...
algo = script.algorithm.threshold()
# Executing...
algo.execute(displayResult=1)

##############

# Algorithm : grouping
# Parameters: operation, outputOption, prefix, childDatasetName, groupingColumns, dataColu
# Creating...
algo = script.algorithm.grouping()
# Executing...
algo.execute(displayResult=1)

##############

# Algorithm : importColumns
# Parameters: fileName, idDataset, idFile,
# Creating...
algo = script.algorithm.importColumns()
# Executing...
algo.execute(displayResult=1)

##############

# Algorithm : labelRows
```

```
# Parameters: label, column,
# Creating...
algo = script.algorithm.labelRows()
# Executing...
algo.execute(displayResult=1)

##############

# Algorithm : KMeans
# Parameters: clusterType, distanceMetric, numClusters, maxIterations, columnIndi
# Creating...
algo = script.algorithm.KMeans()
# Executing...
algo.execute(displayResult=1)

##############

# Algorithm : Hier
# Parameters: clusterType, distanceMetric, linkageRule, columnIndices,
# Creating...
algo = script.algorithm.Hier()
# Executing...
algo.execute(displayResult=1)

##############

# Algorithm : SOM
# Parameters: clusterType, distanceMetric, maxIter, latticeRows, latticeCols, alp
# Creating...
algo = script.algorithm.SOM()
# Executing...
algo.execute(displayResult=1)

##############

# Algorithm : RandomWalk
# Parameters: clusterType, distanceMetric, linkageRule, numIterations, walkDepth
# Creating...
algo = script.algorithm.RandomWalk()
# Executing...
```

```
algo.execute(displayResult=1)

##############

# Algorithm : Eigen
# Parameters: clusterType, distanceMetric, cutoffRatio, columnIndices,
# Creating...
algo = script.algorithm.Eigen()
# Executing...
algo.execute(displayResult=1)

##############

# Algorithm : PcaClustering
# Parameters: clusterType, maxNumClusters, meanShiftToZero, scaleToUnitVariance, columnIn
# Creating...
algo = script.algorithm.PcaClustering()
# Executing...
algo.execute(displayResult=1)

##############

# Algorithm : AxisParallelDTTrain
# Parameters: PruningMethod, GoodnessFunc, LeafImpurity, LeafImpurityType, columnIndices,
# Creating...
algo = script.algorithm.AxisParallelDTTrain()
# Executing...
algo.execute(displayResult=1)

##############

# Algorithm : ObliqueDTTrain
# Parameters: PruningMethod, LeafImpurity, LeafImpurityType, NumIterations, LearningRate,
# Creating...
algo = script.algorithm.ObliqueDTTrain()
# Executing...
algo.execute(displayResult=1)

##############
```

```
# Algorithm : NNTrain
# Parameters: NumNeurons, NumIterations, LearningRate, Momentum, columnIndices, c
# Creating...
algo = script.algorithm.NNTrain()
# Executing...
algo.execute(displayResult=1)

##############

# Algorithm : SVMTrain
# Parameters: kernel, numIterations, cost, ratio, k1, k2, exponent, sigma, colum
# Creating...
algo = script.algorithm.SVMTrain()
# Executing...
algo.execute(displayResult=1)

##############

# Algorithm : AxisParallelDTValidation
# Parameters: PruningMethod, GoodnessFunc, LeafImpurity, LeafImpurityType, NFold
# Creating...
algo = script.algorithm.AxisParallelDTValidation()
# Executing...
algo.execute(displayResult=1)

##############

# Algorithm : ObliqueDTValidation
# Parameters: PruningMethod, LeafImpurity, LeafImpurityType, NumIterations, Learn
# Creating...
algo = script.algorithm.ObliqueDTValidation()
# Executing...
algo.execute(displayResult=1)

##############

# Algorithm : NNValidation
# Parameters: NumNeurons, NumIterations, LearningRate, Momentum, NFold, NumRepeat
# Creating...
algo = script.algorithm.NNValidation()
```

```
# Executing...
algo.execute(displayResult=1)

##############

# Algorithm : SVMValidation
# Parameters: kernel, numIterations, cost, ratio, k1, k2, exponent, sigma, NFold, NumRepea
# Creating...
algo = script.algorithm.SVMValidation()
# Executing...
algo.execute(displayResult=1)

##############

# Algorithm : Classify
# Parameters: model, classLabelColumn,
# Creating...
algo = script.algorithm.Classify()
# Executing...
algo.execute(displayResult=1)

##############

# Algorithm : anovaFeatureSelection
# Parameters: columns,
# Creating...
algo = script.algorithm.anovaFeatureSelection()
# Executing...
algo.execute(displayResult=1)

##############

# Algorithm : kwallisFeatureSelection
# Parameters: columns,
# Creating...
algo = script.algorithm.kwallisFeatureSelection()
# Executing...
algo.execute(displayResult=1)

##############
```

```
# Algorithm : PCA
# Parameters: runOn, pruneBy, columnIndices,
# Creating...
algo = script.algorithm.PCA()
# Executing...
algo.execute(displayResult=1)

#############

# Algorithm : MeanCenter
# Parameters: shouldUseMeanCentring, centerValue, useHouseKeepingOnly, houseKeepi
# Creating...
algo = script.algorithm.MeanCenter()
# Executing...
algo.execute(displayResult=1)

#############

# Algorithm : QuantileNorm
# Parameters: otherparams, columnIndices,
# Creating...
algo = script.algorithm.QuantileNorm()
# Executing...
algo.execute(displayResult=1)

#############
```

18.4.2 Example Scripts to Run Algorithms

```
#********************Example*****************************
#
# run clustering algorithm KMeans on the active dataset
# display the results
#


from script.algorithm import *
```

```
algo = KMeans(numClusters=4)
result = algo.execute()
result.display()

#********************Example*****************************
#
# run SVM Train with specified parameters
# report the overall accuracy
# disply the results
#

from script.algorithm import *

algo = SVMTrain()
algo.kernel = 'Polynomial'
algo.k1 = 0.2
algo.k2 = 1.5
algo.exponent = 3
algo.numIterations = 200

result = algo.execute()

print result.report.overallAccuracy

result.display()
```

18.5 Scripts to Create User Interface in ArrayAssist

Often is may be necessary to create a get inputs for the user and use these imputs to open views, run commands and execute algorithms. **ArrayAssist** provides the a scripting interface to launch user interface elements for the user to provide imputs. The imputs provided can be used to run algorithms or launch views. In this section example scripts are provided that can create such user interfaces in **ArrayAssist**.

```
#A LIST OF ALL UI COMPONENTS CALLABLE BY SCRIPT
```

```
import script
from script.dataset import *
from script.omega import createComponent, showDialog
from javax.swing import *


def textarea(text):
       t = JTextArea(text)
       t.setBackground(JLabel().getBackground())
       return t

#--------------------------------------------------------------------------
#Components appear below
#dropdown
p = createComponent(type="enum", id="name", description="Enumeration",options=["
result=showDialog(p)
print result

#checkbox
p = createComponent(type="boolean", id="name", description="CheckBox")
result=showDialog(p)
print result

#radio
p = createComponent(type="radio", id="name", description="Radio",options=["sdasd"
result=showDialog(p)
print result

#filechooser
p = createComponent(type="file", id="name", description="FileChooser")
result=showDialog(p)
print result

#column choice dropdown
p = createComponent(type="column", id="name", description="SingleColumnChooser",
result=showDialog(p)
print result

#multiple column chooser
p = createComponent(type="columnlist", id="name", description="MultipleColumnCho
```

```
result=showDialog(p)
print result

#textarea
p = createComponent(type="text", id="name", description="TextArea",value="dfdfdffsdfsdfdsf
result=showDialog(p)
print result

#string input, similarly use int and float
p = createComponent(type="string", id="name", description="StringEntry",value="dfdfdffsdfs
result=showDialog(p)
print result

#plain text message
dummytext="""
     Do you like what you see?
   """
p=createComponent(type="ui",  id="name0", description="", component=textarea(dummytext))
result=showDialog(p)
print result

#group components together one below the other
dummytext="""
            Do you like what you see?
   """
p0=createComponent(type="ui",  id="name0", description="", component=textarea(dummytext))
p1 = createComponent(type="string", id="name1", description="String",value="dfdfdffsdfsdfd
p2 = createComponent(type="text", id="name2", description="Text",value="dfdfdffsdfsdfdsf")
p3 = createComponent(type="columnlist", id="name3", description="Columns",dataset=script.p
p4 = createComponent(type="file", id="name4", description="File")
p5 = createComponent(type="radio", id="name5", description="Radio",options=["sdasd","sdasd
panel= createComponent(type="group", id="alltogether", description="Group",components=[p0
result=showDialog(panel)
print result["name0"],result["name1"],result["name2"],result["name3"],result["name4"],resu

#group the same components above but in tabs this time
panel= createComponent(type="tab", id="alltogether", description="Tabs",components=[p0,p1
result=showDialog(panel)
print result["name0"],result["name1"],result["name2"],result["name3"],result["name4"],resu
```

```
#note: YOU CAN GROUP THINGS AND THEN CREATE GROUPS OF GROUPS ETC FOR GOOD FORM DI
```

18.6 Running R Scripts

R scripts can be called from **ArrayAssist** and given access to the dataset in **ArrayAssist** via *Tools* $\longrightarrow$*R Script Editor*. You will need to first set the path to the R executable in the Paths section of *Tools* $\longrightarrow$*Options*, then write or open an R script in this R script editor, and then click on the run button. A failure message below indicates that the R path was not correct. Example R scripts are available in the samples/RScripts subfolder of the installation directory; these show how the **ArrayAssist** dataset can be accessed and sent to R for processing and how the results can be fetched back.

Chapter 19

Table of Key Bindings and Mouse Clicks

All menus and dialogs in **ArrayAssist** adhere to standard conventions on key bindings and mouse clicks. In particular, menus can be invoked using *Alt* keys, dialogs can be disposed using the *Escape* key, etc. On Mac **ArrayAssist** confirms to the standard native mouse clicks.

19.1 Mouse Clicks and their actions

19.1.1 Global Mouse Clicks and their actions

Mouse clicks in different views in **ArrayAssist** perform multiple functions as detailed in the table below:

Mouse Clicks	Action
Left-Click	Brings the view in focus
Left-Click	Selects a row or column or element
Left-Click + Drag	Draws a rectangle and performs selection or zooms into the area as appropriate
Shift + Left-Click	Selects contiguous areas with last selection, where contiguity is well defined
Control + Left Click	Toggles selection in the region
Right-Click	Bring up the context specific menu

Table 19.1: Mouse Clicks and their Action

Mouse Clicks	Action
Shift + Left-Click	Draw Irregular area to select

Table 19.2: Scatter Plot Mouse Clicks

Mouse Clicks	Action
Shift + Left-Click + Move	Rotate the axes of 3D
Shift + Middle-Click + Move up and down	Zoom in and out of 3D
Shift + Right-Click + Move	Translate the axes of 3D

Table 19.3: 3D Mouse Clicks

19.1.2 Some View Specific Mouse Clicks and their Actions

19.2 Key Bindings

These key bindings are effective at all times when the **ArrayAssist** main window is in focus.

19.2.1 Global Key Bindings

Key Binding	Action
Ctrl-O	Open new dataset from File
Ctrl-S	Save current dataset to File
Ctrl-W	Close current dataset
Ctrl-X	Quit **ArrayAssist**
Ctrl-D	Open Dataset Properties
Ctrl-R	Open View Properties
Ctrl-L	Open Log Window
Ctrl-A	Open Lasso View
Ctrl-M	Launch Memory Monitor
Ctrl-E	Open Script Editor
Ctrl-C	Copy View to System Clipboard
Ctrl-V	Paste from System Clipboard
Ctrl-P	Print

Table 19.4: Global Key Bindings

19.2.2 View Specific Key Bindings

These key bindings apply only to specific views as described below.

Key Binding	Action
Ctrl-C	Copy selected columns to buffer
Ctrl-X	Cut selected columns to buffer
Ctrl-V	Paste columns in buffer to spreadsheet

Table 19.5: Spreadsheet Key Bindings

Key Binding	Action
x	Activate X-Axis dropdown list
y	Activate Y-Axis dropdown list

Table 19.6: Scatter Plot Key Bindings

Key Binding	Action
c	Activate Channel dropdown list

Table 19.7: Histogram Key Bindings

Bibliography

[1] Rafael. A. Irizarry, Benjamin M. Bolstad, Francois Collin, Leslie M. Cope, Bridget Hobbs and Terence P. Speed (2003), Summaries of Affymetrix GeneChip probe level data Nucleic Acids Research 31(4):e15

[2] Irizarry, RA, Hobbs, B, Collin, F, Beazer-Barclay, YD, Antonellis, KJ, Scherf, U, Speed, TP (2003) Exploration, Normalization, and Summaries of High Density Oligonucleotide Array Probe Level Data. Biostatistics .Vol. 4, Number 2: 249-264 [Abstract, PDF, PS, Complementary Color Figures-PDF, Software]

[3] Bolstad, B.M., Irizarry R. A., Astrand M., and Speed, T.P. (2003), A Comparison of Normalization Methods for High Density Oligonucleotide Array Data Based on Bias and Variance. Bioinformatics 19(2):185-193 Supplemental information

[4] Hubbell, E., et al. Robust estimators for expression analysis. Bioinformatics. 2002, 18(12):1585-92

[5] Hubbell, E., Designing Estimators for Low Level Expression Analysis. http://mbi.osu.edu/2004/ws1abstracts.html

[6] Li, C. and W.H. Wong (2001) Model based analysis of oligonucleotide arrays: Expression index computation and outlier detection, PNAS Vol. 98: 31-36.

[7] Zhijin Wu, Rafael A. Irizarry, Robert Gentleman, Francisco Martinez Murillo, and Forrest Spencer, A Model Based Background Adjustment for Oligonucleotide Expression Arrays (May 28, 2004). Johns Hopkins University, Dept. of Biostatistics Working Papers. Working Paper 1.

[8] Affymetrix Latin Square Data. http://www.affymetrix.com/support/technical/sample_data/datasets.affx

[9] GeneLogic Spike In Study. http://www.genelogic.com/media/studies/spikein.cfm

[10] Comparison of Probe Level Algorithms. http://affycomp.biostat.jhsph.edu

[11] Bolstad BM, Irizarry RA, Astrand M, Speed TP: A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. Bioinformatics, 19, 2, 185–193, 2003.

[12] Hill AA, Brown EL, Whitley MZ, Tucker-Kellog G, Hunter CP, Slonim DK: Evaluation of normalization procedures for Oligonucleotide array data based on spiked cRNA controls, Genome Biology, 2, 0055.1-0055.13, 2001.

[13] Hoffmann R, Seidl T, Dugas M: Profound effect of normalization on detection of differentially expressed genes in oligonucleotide microarray data analysis, Genome Biology. 3(7), 0033.1-0033.11, 2002.

[14] Li C, Wong WH: Model-based analysis of oligonucleotide arrays: expression index computation and outlier detection. Proc Natl Acad Sci USA. 98, 31-36, 2000.

[15] Li C, Wong WH: Model-based analysis of oligonucleotide arrays: model validation, design issues and standard error application, Genome Biology. 2(8), 0032.1-0032.11, 2001.

[16] Irizarry, RA, Hobbs B, Collin F, Beazer-Barclay YD, Antonellis KJ, Scherf U, Speed T.P: Exploration, normalization and summaries of high density oligonucleotide array probe level data. Biostatistics. 4(2), 249-264, 2003.

[17] The Bioconductor Webpage. http://www.bioconductor.org.

Validation of Sequence-Optimized 70 Base Oligonucleotides for Use on DNA Microarrays, Poster at http://www.operon.com/arrays/poster.php.

[18] DChip: The DNA Chip Analyzer. http://www.biostat.harvard.edu/complab/dchip.

[19] Gene Logic Latin Square Data. http://qolotus02.genelogic.com.

[20] The Lowess method. http://www.itl.nist.gov/div898/handbook/pmd/section1/pmd144.htm.

[21] Strand Genomics **ArrayAssist**. http://avadis.strandgenomics.com

[22] T. Speed: Always log spot intensities and ratios, Speed Group Microarray Page. http://stat-www.berkeley.edu/users/terry/zarray/Html/log.html.

[23] Statistical Algorithms Description Document, Affymetrix Inc. http://www.affymetrix.com/support/technical/whitepapers/sadd_whitepaper.pdf.

[24] Benjamini B, Hochberg Y: Controlling the false discovery rate: a practical and powerful approach to multiple testing. J. R. Statist. Soc. B. 57, 289-300, 1995.

[25] Dudoit S, Yang H, Callow MJ, Speed TP: Statistical Methods for identifying genes with differential expression in replicated cDNA experiments, Stat. Sin. 12, 1, 11-139, 2000.

[26] Glantz S: Primer of Biostatistics, 5th edition, McGraw-Hill, 2002.

[27] Westfall PH, Young SS: Resampling based multiple testing. John Wiley and Sons. New York, 1993.